

Eilenberg–Moore Monoids and Backtracking Monad Transformers

Maciej Piróg

Department of Computer Science
KU Leuven, Belgium

maciej.pirog@cs.kuleuven.be

We develop an algebraic underpinning of backtracking monad transformers in the general setting of monoidal categories. As our main technical device, we introduce *Eilenberg–Moore monoids*, which combine monoids with algebras for strong monads. We show that Eilenberg–Moore monoids coincide with algebras for the list monad transformer (‘done right’) known from Haskell libraries. From this, we obtain a number of results, including the facts that the list monad transformer is indeed a monad, a transformer, and an instance of the `MonadPlus` class. Finally, we construct an Eilenberg–Moore monoid of endomorphisms, which, via the codensity monad construction, yields a continuation-based implementation à la Hinze.

1 Introduction

In monadic functional programming, the most straightforward approach to backtracking is realised by the list monad [9, 30]. More advanced structures are used for efficient implementation or for combining backtracking with other computational effects [13, 20]. This paper is concerned with a category-theoretic explanation of such more advanced structures.

Historically, there has been some dispute over the ‘correct’ definition of the monad transformer associated with the list monad. First, a popular Haskell library `mtl`¹ proposed the following definition (in a pseudo-Haskell syntax):

```
type ListT m a = m [a]
```

The idea behind this type is that each computation first performs some effects in `m`, and then returns a list of results. This structure is not entirely satisfactory. One problem with it is strictly mathematical: the list monad transformer defined in this way is not really a monad transformer, since it does not satisfy the required equations in general (it does when the transformed monad `m` is commutative). A conceptual disadvantage is that it does not fully reflect the backtracking aspect of list computations: if we want to look only for the first result, we are not necessarily interested in the effects associated with the subsequent results of the computation, while in this implementation all `m`-effects are performed immediately.

In this paper, we consider an alternative definition of the list monad transformer, known as ‘`ListT` done right’, which is provided by the Haskell packages `list-t`, `List`, and `pipes`. It can be implemented as follows:

```
data ListT m a = ListT (m (Maybe (a, ListT m a)))
```

The idea is that a value of this type is a list in which each tail is guarded by `m`. Intuitively, if we want to extract the n -th result from the computation, we first have to perform the effects guarding the elements 1

¹Documentation for each Haskell package is available online at <http://hackage.haskell.org/package/PACKAGE-NAME>.

to n . Jaskelioff and Moggi [18] propose an equational presentation of this monad, on which we expand and generalise in this paper (in particular, we abstract to general monoidal categories).

Another approach to backtracking transformers—taken, for example, by Hinze [13] and Kiselyov *et al.* [20]—is to employ continuations. Hinze proposes the following monad, which uses a pair of success and failure continuations:

```
type Backtr m a = forall x. (a -> m x -> m x) -> m x -> m x
```

Hinze derives this monad using Hughes’s [15] ‘context-passing’ technique. As one of our contributions, we relate it to the list monad transformer using the codensity monad construction [14].

In this paper, we model structures like those mentioned above in a monoidal category \mathcal{C} (or, when the continuation-based structures are involved, a closed monoidal category), starting with an algebraic specification of backtracking. As our point of departure, we define *Eilenberg–Moore monoids*, which combine the theory of a strong monad (the ‘transformed’ monad), the theory of monoids (which gives us choice and failure), and a coherence condition that specifies the order in which computations are executed. We construct the free monad induced by Eilenberg–Moore monoids, which coincides with the list monad transformer ‘done right’, and introduce a ‘Cayley representation’, which gives us an isomorphic continuation-based implementation. In detail, our contributions are the following:

- In Section 3, given a strong monad M , we introduce *Eilenberg–Moore M -monoids*. They are tuples $\langle A, a, m, u \rangle$, where $\langle A, a \rangle$ is an Eilenberg–Moore M -algebra, and $\langle A, m, u \rangle$ is a monoid in the ambient monoidal category, that satisfy a coherence condition bringing together the algebra structure, the monoid structure, and the strength of the monad. Then, assuming certain algebraically free monads exist, we construct free Eilenberg–Moore monoids. The monad UF induced by the free–underlying adjunction $F \dashv U$ is the list monad transformer applied to M .
- In Section 4, we show that the category of Eilenberg–Moore M -monoids is isomorphic to the category of Eilenberg–Moore algebras for the list monad transformer applied to M . As an application of this result, we employ a correspondence between monad morphisms and functors between Eilenberg–Moore categories to obtain a monad transformer structure.
- In Section 5, assuming that the ambient category is closed, we introduce an Eilenberg–Moore monoid of endomorphisms $MA \Rightarrow MA$. We prove that it is a Cayley representation of a certain subcategory of Eilenberg–Moore monoids. Since this subcategory contains all free Eilenberg–Moore monoids, this gives us, via the codensity monad construction [14], a continuation-based monad transformer isomorphic to the list monad transformer. It turns out to coincide with the mentioned construction introduced by Hinze [13].
- In Section 6, to show that the used techniques can be applied more universally, we revisit the transformer for commutative monads $m \ [a]$. We characterise its algebras, which turn out to form a subcategory of Eilenberg–Moore M -monoids for a commutative monad M . We introduce a Cayley representation $A \Rightarrow MA$ of a sufficient subcategory of these monoids, and obtain a novel continuation-based implementation.

Thus, we relate a number of existing constructions and introduce one new ‘commutative monad’ transformer. We take a high-level approach—for example, we use the resumption monad and its universal properties to describe free Eilenberg–Moore monoids, which liberates us from tedious proofs by structural induction. The seemingly arbitrary structure related to the list monad transformer (like the `lift` monad morphism or the instance of the `MonadPlus` class) is simply a corollary of the general results, and is not only correct by construction, but also recognised as canonically related to the free–underlying adjunction $F \dashv U$. The category-theoretic approach allows us to split the tackled constructions

into more fine-grained pieces; for example, the role of monadic strength becomes apparent, although in Haskell, where all monads are canonically strong, it is usually used implicitly, obfuscating equational reasoning.

2 Background

We denote categories by $\mathcal{C}, \mathcal{D}, \dots$. We work in a monoidal category $\langle \mathcal{C}, \otimes, I \rangle$ (we additionally assume that it is closed in Section 5, and symmetric closed in Section 6). We use the symbol \cong for the structural natural isomorphisms of monoidal categories (instead of the more traditional λ, ρ , and α). We always give types explicitly, so no confusion should arise. Also, we skip the subscripts in natural transformations when the component is obvious from the type (so, for example, we write $MMA \xrightarrow{\mu} MA$ instead of $MMA \xrightarrow{\mu_A} MA$).

2.1 Monoids in monoidal categories

A *monoid* in \mathcal{C} is a triple $\langle A, A \otimes A \xrightarrow{m} A, I \xrightarrow{u} A \rangle$, where m and u are called the *multiplication* and the *unit* respectively, such that the following diagrams commute:

$$\begin{array}{ccc} I \otimes A & \xrightarrow{u \otimes \text{id}} & A \otimes A \\ & \searrow \cong & \downarrow m \\ & & A \end{array} \quad \begin{array}{ccc} A \otimes I & \xleftarrow{\text{id} \otimes u} & A \otimes A \\ & \swarrow \cong & \downarrow m \\ & & A \end{array}$$

$$\begin{array}{ccccc} (A \otimes A) \otimes A & \xrightarrow{m \otimes \text{id}} & A \otimes A & & \\ \downarrow \cong & & \downarrow m & & \\ A \otimes (A \otimes A) & \xrightarrow{\text{id} \otimes m} & A \otimes A & \xrightarrow{m} & A \end{array}$$

A morphism between monoids $\langle A, m^A, u^A \rangle$ and $\langle B, m^B, u^B \rangle$ is a morphism $h : A \rightarrow B$ in \mathcal{C} such that $h \cdot m^A = m^B \cdot (h \otimes h)$ and $h \cdot u^A = u^B$. The category of monoids and morphisms between monoids is called **Mon**. If the obvious forgetful functor $U_{\mathbf{Mon}} : \mathbf{Mon} \rightarrow \mathcal{C}$ has a left adjoint $F_{\mathbf{Mon}}$, the induced monad is a generalisation of the list monad.

2.2 Monads and strength

To set the notation, we give a number of basic definitions of monads and some related concepts. A *monad* on \mathcal{C} is a triple $\langle M, MM \xrightarrow{\mu} M, \text{Id} \xrightarrow{\eta} M \rangle$ such that it is a monoid in the monoidal category of endofunctors on \mathcal{C} with natural transformations as morphisms, and the monoidal tensor given by composition of endofunctors. We denote a monad simply by its carrier M , and the monadic structure, that is, the multiplication and the unit, are always denoted μ and η respectively. If there are a number of monads in the context, we sometimes put the name of the monad in superscript, for example μ^M and η^M . The category of monads on \mathcal{C} and monad morphisms (that is, morphism between appropriate monoids) is denoted **Mnd**.

An *Eilenberg–Moore algebra* (or simply an *algebra*) for a monad M on \mathcal{C} is a pair $\langle A, MA \xrightarrow{a} A \rangle$, where A is an object in \mathcal{C} , while a is a morphism such that $a \cdot Ma = a \cdot \mu_A$ and $a \cdot \eta_A = \text{id}_A$. If $a : MA \rightarrow A$ is a morphism such that the pair $\langle A, a \rangle$ is an Eilenberg–Moore algebra, we say that a has the Eilenberg–Moore property. A morphism between two Eilenberg–Moore algebras $\langle A, a \rangle$ and $\langle B, b \rangle$ is a morphism $h : A \rightarrow B$ such that $b \cdot Mh = h \cdot a$. The category of such algebras and such morphisms is called the *Eilenberg–Moore category* of M .

For a monad M on \mathcal{C} there is an adjunction between \mathcal{C} and the Eilenberg–Moore category of M given on objects as $A \mapsto \langle MA, \mu_A \rangle$ in one direction and $\langle A, a \rangle \mapsto A$ in the other direction. For every other adjunction $L \dashv R : \mathcal{C} \rightarrow \mathcal{D}$ that induces M , there exists a functor (called the *comparison functor*) from \mathcal{D} to the Eilenberg–Moore category of M . It is defined on objects as

$$X \mapsto \langle RX, MRX \xrightarrow{\cong} RLRX \xrightarrow{R\varepsilon} RX \rangle,$$

where ε is the counit of the adjunction $L \dashv R$. If the comparison functor is an isomorphism, we say that the adjunction $L \dashv R$ is *strictly monadic*. An example of a strictly monadic adjunction is $F_{\mathbf{Mon}} \dashv U_{\mathbf{Mon}}$. This entails that the Eilenberg–Moore category of the list monad is isomorphic to the category of monoids.

An endofunctor G is *strong* if it is equipped with a transformation $\tau : GA \otimes B \rightarrow G(A \otimes B)$ (called a *strength* of G) natural in A and B such that the following diagrams commute:

$$\begin{array}{ccc} GA \otimes I & \xrightarrow{\tau} & G(A \otimes I) \\ & \searrow \cong & \downarrow \cong \\ & & GA \end{array} \quad \begin{array}{ccc} GA \otimes (B \otimes C) & \xrightarrow{\tau} & G(A \otimes (B \otimes C)) \\ \downarrow \cong & & \downarrow \cong \\ (GA \otimes B) \otimes C & \xrightarrow{\tau \otimes \text{id}} & G(A \otimes B) \otimes C \xrightarrow{\tau} G((A \otimes B) \otimes C) \end{array}$$

A monad M is *strong* if it is strong as an endofunctor, and additionally the following diagrams commute:

$$\begin{array}{ccc} A \otimes B & \xrightarrow{\eta \otimes \text{id}} & MA \otimes B \\ & \searrow \eta & \downarrow \tau \\ & & M(A \otimes B) \end{array} \quad \begin{array}{ccc} MMA \otimes B & \xrightarrow{\mu \otimes \text{id}} & MA \otimes B \\ \downarrow \tau & & \downarrow \tau \\ M(MA \otimes B) & \xrightarrow{M\tau} & MM(A \otimes B) \xrightarrow{\mu} M(A \otimes B) \end{array}$$

While a monad can have more than one strength, in **Set** (and in Haskell), every monad is equipped with a canonical strength given by $\langle m, b \rangle \mapsto (M(\lambda a. \langle a, b \rangle))(m)$. A strong monad on \mathcal{C} can be turned into a monoid in \mathcal{C} :

Theorem 1 (Wolff [32]). *Let M be a strong monad on \mathcal{C} . Then, the tuple $\langle MI, \mathfrak{m}, \mathfrak{u} \rangle$, where*

$$\mathfrak{m} = \left(MI \otimes MI \xrightarrow{\tau} M(I \otimes MI) \xrightarrow{\cong} MMI \xrightarrow{\mu} MI \right) \text{ and } \mathfrak{u} = \left(I \xrightarrow{\eta} MI \right),$$

is a monoid.

2.3 Strongly generated algebraically free monads

Given an endofunctor G on \mathcal{C} , consider the category of G -algebras, that is, pairs $\langle A, GA \xrightarrow{a} A \rangle$, and morphisms $\langle A, a \rangle \rightarrow \langle B, b \rangle$ given by \mathcal{C} -morphisms $h : A \rightarrow B$ such that $b \cdot Gh = h \cdot a$. If the obvious forgetful functor from the category of G -algebras to \mathcal{C} has a left adjoint, we denote the free algebra generated by an object A as $\langle G^*A, GG^*A \xrightarrow{\text{cons}} G^*A \rangle$, and call the induced monad G^* the *algebraically free monad* generated by G . This adjunction is strictly monadic, which entails that the category of G -algebras is the Eilenberg–Moore category of G^* . We denote the monadic structure of the algebraically free monad as $\mu^{\mathcal{F}}$ and $\eta^{\mathcal{F}}$. The freeness property of $\langle G^*A, \text{cons} \rangle$ can be described as follows, where the morphism emb is given by $GA \xrightarrow{G\eta^{\mathcal{F}}} GG^*A \xrightarrow{\text{cons}} G^*A$.

Theorem 2. For a morphism $GA \xrightarrow{g} A$, there exists a unique morphism between algebras $\llbracket g \rrbracket : \langle G^*A, \text{cons} \rangle \rightarrow \langle A, g \rangle$ such that the following diagram commutes:

$$\begin{array}{ccc} GA & \xrightarrow{\text{emb}} & G^*A \\ & \searrow g & \downarrow \llbracket g \rrbracket \\ & & A \end{array}$$

If \mathcal{C} has coproducts, the carrier of the free algebra $\langle G^*A, \text{cons} \rangle$ can be given as the carrier of the initial $(G(-) + A)$ -algebra (if it exists), that is, $G^*A = \mu X. GX + A$. For endofunctors G and H , if both G^* and H^* exist, every natural transformation $h : G \rightarrow H$ induces a monad morphism $h^* : G^* \rightarrow H^*$ given as $G^*A \xrightarrow{G^*\eta} G^*H^*A \xrightarrow{\llbracket \text{cons} \cdot h \rrbracket} H^*A$. Note that we use the notation $(-)^*$ as if it was a functor, although in general not all endofunctors induce algebraically free monads. If G is strong, we define in what way G^* can inherit the strength of G in a coherent fashion:

Definition 3. Let G be a strong endofunctor such that G^* exists. We say that G^* is *strongly generated* if for all morphisms $f : A \otimes B \rightarrow C$ and a G -algebra $\langle C, c \rangle$, there exists a unique morphism \hat{f} that makes the following diagram commute:

$$\begin{array}{ccccc} GG^*A \otimes B & \xrightarrow{\tau} & G(G^*A \otimes B) & \xrightarrow{G\hat{f}} & GC \\ \downarrow \text{cons} \otimes \text{id} & & & & \downarrow c \\ G^*A \otimes B & \xrightarrow{\hat{f}} & & & C \\ \uparrow \eta^{\mathcal{F}} \otimes \text{id} & & & & \uparrow f \\ A \otimes B & & & & \end{array}$$

A strongly generated monad is strong with the strength given by $\hat{\eta}$ for $\eta : A \otimes B \rightarrow M(A \otimes B)$. Moreover, for a natural transformation $h : G \rightarrow H$, the monad morphism $h^* : G^* \rightarrow H^*$ preserves strength, that is, $h^* \cdot \tau = \tau \cdot (h^* \otimes \text{id})$. If the category \mathcal{C} is closed (for example, **Set**), all algebraically free monads generated by strong endofunctors are strongly generated (see Fiore [11, Theorem 4.4]). In this paper, we assume that all the algebraically free monads that we deal with are strongly generated.

2.4 The resumption monad

Another construction that we use is the *resumption monad* introduced by Moggi [24], also known as the *free monad transformer*. Given a monad M and an endofunctor G , it is given as the composition $M(GM)^*$ if $(GM)^*$ exists. In the case of the free monad given by initial algebras, it becomes $A \mapsto M(\mu X. GMX + A)$. Using the rolling lemma [5], it is isomorphic to $A \mapsto \mu X. M(GX + A)$.

We notice that the endofunctor part of the list monad transformer that we work with in this paper can be given by initial algebras $A \mapsto \mu X. M((A \otimes X) + I) \cong M(A \otimes M(-))^*I$, which are similar in shape to the resumption monad for the endofunctor $(A \otimes -)$ applied to the object I (and this is how it is implemented in the Haskell package `pipes`). Although the monadic structure of LMT_M is not given directly by the monadic structure of the resumption monad, the two are related.

Hyland, Plotkin, and Power [16] show two important properties of the resumption monad. The first one is that it is induced by a distributive law $\lambda : (GM)^*M \rightarrow M(GM)^*$. Thus, its monadic structure can

be defined as $\mu^{\mathcal{R}} = \mu^M \mu^{\mathcal{F}} \cdot M\lambda(GM)^*$ and $\eta^{\mathcal{R}} = \eta^M \eta^{\mathcal{F}}$. Moreover, a natural transformation $h : G \rightarrow H$ induces a monad morphism $M(hM)^* : M(GM)^* \rightarrow M(HM)^*$. In general, the composition of two strong endofunctors is strong, and the composition of two strong monads via a distributive law is strong. Thus, if M and G are strong, and $(GM)^*$ is strongly generated, the resumption monad $M(GM)^*$ is strong.

The other important property of the resumption monad is that $M(GM)^*$ is the coproduct in **Mnd** of M and G^* . A classical result by Kelly [19] states that the coproduct in **Mnd** of two monads M and T is always given by the free–underlying adjunction between the base category and the category of tuples $\langle A, MA \xrightarrow{m} A, TA \xrightarrow{t} A \rangle$, where both m and t have the Eilenberg–Moore property. Moreover, this adjunction is strictly monadic. Thus, using the correspondence between Eilenberg–Moore algebras for algebraically free monads and algebras for endofunctors, Hyland, Plotkin, and Power describe the Eilenberg–Moore category of the resumption monad for an endofunctor G and a monad M as consisting of tuples $\langle A, MA \xrightarrow{a} A, GA \xrightarrow{g} A \rangle$ such that a has the Eilenberg–Moore property. Adapting Kelly’s result, the freeness property of the resumption monad can be stated as follows, where the natural transformation INF (injection of the functor) is defined as:

$$\text{INF} = \left(G \xrightarrow{G\eta^M} GM \xrightarrow{\text{emb}} (GM)^* \xrightarrow{\eta^M} M(GM)^* \right)$$

Theorem 4. *Given an Eilenberg–Moore algebra $\langle B, MB \xrightarrow{b} B \rangle$, an algebra $\langle B, GB \xrightarrow{g} B \rangle$, and a morphism $h : A \rightarrow B$, there exists a unique morphism $\langle\langle b, g, h \rangle\rangle : M(GM)^*A \rightarrow B$ such that the following diagrams commute:*

$$\begin{array}{ccccccc} MM(GM)^*A & \xrightarrow{\mu^M} & M(GM)^*A & \xleftarrow{\mu^{\mathcal{R}}} & M(GM)^*M(GM)^*A & \xleftarrow{\text{INF}} & GM(GM)^*A & & A & \xrightarrow{\eta^{\mathcal{R}}} & M(GM)^*A \\ \downarrow M\langle\langle b, g, h \rangle\rangle & & \downarrow \langle\langle b, g, h \rangle\rangle & & & & \downarrow G\langle\langle b, g, h \rangle\rangle & & \searrow h & & \downarrow \langle\langle b, g, h \rangle\rangle \\ MB & \xrightarrow{b} & B & \xleftarrow{g} & GB & & & & & & B \end{array}$$

Morphisms $\langle\langle -, -, - \rangle\rangle$ enjoy some equational properties that we find useful in the remainder of this paper:

Lemma 5. *Given b , g , and h as in the previous theorem, the following hold:*

1. *The morphism $\langle\langle b, g, h \rangle\rangle$ can be expressed as $M(GM)^*A \xrightarrow{M(GM)^*h} M(GM)^*B \xrightarrow{\langle\langle b, g, \text{id} \rangle\rangle} B$.*
2. *The morphism $\langle\langle b, g, \text{id} \rangle\rangle$ has the Eilenberg–Moore property, and it is equal to $M(GM)^*B \xrightarrow{M\llbracket g \cdot Gb \rrbracket} MB \xrightarrow{b} B$.*
3. *The morphism $GA \xrightarrow{\text{INF}} M(GM)^*A \xrightarrow{\langle\langle b, g, h \rangle\rangle} B$ is equal to $GA \xrightarrow{Gh} GB \xrightarrow{g} B$.*

3 Eilenberg–Moore monoids

In this section, we introduce Eilenberg–Moore monoids, which serve as an algebraic specification of backtracking combined with other effects. Then, we construct free Eilenberg–Moore monoids, which induce the list monad transformer.

Definition 6. Let M be a strong monad on a monoidal category \mathcal{C} . An *Eilenberg–Moore M -monoid* is a tuple

$$\langle A, MA \xrightarrow{a} A, A \otimes A \xrightarrow{m} A, I \xrightarrow{u} A \rangle$$

such that the following hold:

1. $\langle A, a \rangle$ is an Eilenberg–Moore M -algebra,
2. $\langle A, m, u \rangle$ is a monoid,
3. coherence: the following diagram commutes:

$$\begin{array}{ccccc}
 MA \otimes A & \xrightarrow{a \otimes \text{id}} & A \otimes A & & \\
 \downarrow \tau & & \downarrow m & & \\
 M(A \otimes A) & \xrightarrow{Mm} & MA & \xrightarrow{a} & A
 \end{array}$$

A morphism between two Eilenberg–Moore M -monoids is a morphism in \mathcal{C} that is both a morphism between M -algebras and between monoids. We call the category of Eilenberg–Moore M -monoids and such morphisms **EMMon** $_M$.

Example 7. Let \mathcal{C} be **Set**. Let $M = G^*$ be the free monad generated by the functor $GX = X \times X$, that is, G^* is the free monad of the theory of a single binary operation. Consider the tuple $\langle \mathbb{N}, \llbracket (+) \rrbracket : G^*\mathbb{N} \rightarrow \mathbb{N}, (*): \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}, \lambda x.1 : I \rightarrow \mathbb{N} \rangle$, in which we interpret the monad operation as addition, while the monoid is given by natural numbers with multiplication. It is an Eilenberg–Moore monoid. In this case, the coherence condition amounts to the right-distributivity of multiplication over addition.

Although to the author’s best knowledge Eilenberg–Moore M -monoid is a new concept, it is an obvious generalisation of the concept of F -monoid:

Definition 8 (Fiore, Plotkin, and Turi [12]). *F-monoids* are similar to Eilenberg–Moore monoids, but we drop the condition (1) from Definition 6 and the assumption that the endofunctor is a monad (that is, it is merely a strong endofunctor).

We also need the following technical lemma, which relates F -monoids to Eilenberg–Moore monoids for free monads. Note that it could be used for a simple proof that the tuple from Example 7 is an Eilenberg–Moore monoid.

Lemma 9. *Let G be a strong endofunctor that strongly generates a free monad G^* . Then, each G -monoid $\langle A, GA \xrightarrow{g} A, A \otimes A \xrightarrow{m} A, I \xrightarrow{u} A \rangle$ gives rise to an Eilenberg–Moore G^* -monoid $\langle A, G^*A \xrightarrow{\llbracket g \rrbracket} A, m, u \rangle$.*

From the perspective of algebraic effects (see Hyland and Power [17]), Eilenberg–Moore M -monoids can be seen as the right-distributive tensor of the theory of monoids over the theory of the monad M . Each Eilenberg–Moore monoid consists of a monoid and an interpretation of operations provided by M , while, denoting the monoid multiplication as \vee , for an n -ary M -operation f , the coherence diagram becomes the following equation:

$$f(x_1, \dots, x_n) \vee y = f(x_1 \vee y, \dots, x_n \vee y)$$

Intuitively, it states that when making a choice between two values, the effects in the left-hand argument are always executed first. This differentiates backtracking from plain nondeterminism, where no order of execution is imposed. Note that the list monad is used for backtracking usually in lazy languages, where laziness is the effect that defines the order in which elements arrive. In eager languages, backtracking can be implemented using *lazy lists*, in which laziness is an explicit effect. Indeed, the type of lazy lists is an instance of the list monad transformer (see Section 7).

3.1 Free Eilenberg–Moore monoids

Now, we describe free Eilenberg–Moore monoids, which give us the monadic structure of the list monad transformer. First, we need an auxiliary definition:

Definition 10. For a monad M and a \mathcal{C} -object A , we define the functor $(A \ltimes M)X = A \otimes MX$.

Note that each morphism $f : A \rightarrow B$ induces a natural transformation $f \ltimes M : A \ltimes M \rightarrow B \ltimes M$. If M is strong, the functor $A \ltimes M$ is strong via $(A \otimes MB) \otimes C \xrightarrow{\cong} A \otimes (MB \otimes C) \xrightarrow{\text{id} \otimes \tau} A \otimes M(B \otimes C)$. If the monad $(A \ltimes M)^*$ exists and is strongly generated, we denote the resulting strength as $\tilde{\tau} : (A \ltimes M)^* B \otimes C \rightarrow (A \ltimes M)^*(B \otimes C)$. Our main result follows.

Theorem 11. *Let M be a strong monad. Assume that there exists a strongly generated algebraically free monad $(A \ltimes M)^*$ for all objects A . Then, the obvious forgetful functor $U : \mathbf{EMMon}_M \rightarrow \mathcal{C}$ has a left adjoint $F : \mathcal{C} \rightarrow \mathbf{EMMon}_M$ given as follows:*

$$\begin{aligned} FA &= \langle M(A \ltimes M)^* I, \mu^M, \mathfrak{m}, \mathfrak{u} \rangle \\ F(f : A \rightarrow B) &= M(f \ltimes M)^* I, \end{aligned}$$

where $\langle M(A \ltimes M)^* I, \mathfrak{m}, \mathfrak{u} \rangle$ is the monoid induced by the resumption monad as described in Theorem 1.

In detail, the associated natural isomorphism $[-] : \mathbf{EMMon}_M(FA, \langle B, b, m^B, u^B \rangle) \cong \mathcal{C}(A, B) : [-]$ is defined as follows. Let $f : FA \rightarrow \langle B, b, m^B, u^B \rangle$ be a morphism between Eilenberg–Moore M -monoids. The \mathcal{C} -morphism $[f] : A \rightarrow B$ is defined as:

$$A \xrightarrow{\cong} A \otimes I \xrightarrow{\text{INF}} M(A \ltimes M)^* I \xrightarrow{f} B$$

In the other direction, let $g : A \rightarrow U\langle B, b, m^B, u^B \rangle$ be a \mathcal{C} -morphism. Then, the \mathbf{EMMon}_M -morphism $[g] : FA \rightarrow \langle B, b, m^B, u^B \rangle$ is defined as:

$$M(A \ltimes M)^* I \xrightarrow{\langle MB \xrightarrow{b} B, A \otimes B \xrightarrow{g \otimes \text{id}} B \otimes B \xrightarrow{m^B} B, I \xrightarrow{u^B} B \rangle} B$$

Definition 12. We call the monad UF induced by the adjunction above the *list monad transformer* and denote it as LMT_M .

To get a more direct definition of the monadic structure of LMT_M , let $\varepsilon_{\langle A, a, m^A, u^A \rangle} = [A \xrightarrow{\text{id}} A]$ be the counit of the adjunction. The monad multiplication is thus given as follows:

$$(UFUF \xrightarrow{U\varepsilon F} UF) = (M(M(A \ltimes M)^* \ltimes M)^* I \xrightarrow{[\text{id}] = \langle \mu^M, \mathfrak{m}, \mathfrak{u} \rangle} M(A \ltimes M)^* I)$$

The unit of LMT_M is given as: $[\text{id}] = (A \xrightarrow{\cong} A \otimes I \xrightarrow{\text{INF}} M(A \ltimes M)^* I)$.

We can also verify that the morphisms \mathfrak{m} and \mathfrak{u} form a *MonadPlus* structure [28]. They obviously form a monoid, so it is left to verify two additional laws: left distributivity and left zero (or, in the language of Plotkin and Power [27], that \mathfrak{m} and \mathfrak{u} are *algebraic*). The desired laws are simply the preservation of the multiplication and the unit by $U\varepsilon F$, which follows from the fact that εF is a morphism between Eilenberg–Moore monoids.

4 Algebras for the list monad transformer

The previous section shows a construction of an adjunction $F \dashv U$ that gives rise to the monad LMT_M , but it is not an ordinary adjunction: we show that it is strictly monadic. We use this fact to construct some monad morphisms.

Theorem 13. *The adjunction $F \dashv U$ is strictly monadic. This entails that the category \mathbf{EMon}_M is isomorphic to the category of Eilenberg–Moore algebras of the monad LMT_M .*

As an application of Theorem 13, we show that LMT is indeed a monad transformer, that is, we construct a monad morphism from a monad M to LMT_M . Instead of defining the morphism directly and mundanely verifying the necessary properties, we utilise the following theorem (see Barr and Wells [6, Ch. 3, Theorem 6.3]):

Theorem 14. *Let \mathbf{EM} be a category in which objects are Eilenberg–Moore categories of monads on \mathcal{C} , while morphisms are carrier-preserving functors (that is, functors that commute with the forgetful functors from Eilenberg–Moore categories to \mathcal{C}). Then, there exists an isomorphism $\mathbf{Mnd} \cong \mathbf{EM}^{\text{op}}$, where \mathbf{Mnd} is the category of monads on \mathcal{C} and monad morphisms.*

In detail, given two monads T and M , consider a carrier-preserving functor F , and let $\langle TA, MTA \xrightarrow{a} TA \rangle = F \langle TA, \mu^T \rangle$. Now, the monad morphism corresponding to F is given for an object A as $MA \xrightarrow{M\eta^T} MTA \xrightarrow{Ma} TA$.

There exist obvious forgetful functors from \mathbf{EMon}_M to the category of monoids and to the category of algebras for M . These give us two monad morphisms: from the list monad and from M respectively. The latter is the desired `lift` operation of monad transformers. Following the description above, it is given as follows:

$$\left(MA \xrightarrow{M\eta^{\text{LMT}_M}} MM(A \ltimes M)^* I \xrightarrow{\mu^M} M(A \ltimes M)^* I \right) = \left(MA \xrightarrow{M\cong} M(A \otimes I) \xrightarrow{M\text{emb}} M(A \ltimes M)^* I \right)$$

Moreover, there exists a forgetful functor from \mathbf{EMon}_M to the Eilenberg–Moore category of the resumption monad generated by the endofunctor $A \mapsto A \otimes A$. This forgetful functor is given as $\langle A, a, m, u \rangle \mapsto \langle A, a, m \rangle$, and it induces a monad morphism from the resumption monad to LMT_M , which flattens the tree structure into a list.

5 Continuation-based implementation

In this section, we deal with a continuation-based backtracking monad transformer à la Hinze [13] mentioned in the introduction. We derive it from the list monad transformer using the codensity monad construction, thus automatically obtaining that the two monads are isomorphic. First, we discuss some background on closed monoidal categories, which we need to model continuations.

5.1 Background: closed monoidal categories

A monoidal category $\langle \mathcal{C}, \otimes, I \rangle$ is *closed* if for all \mathcal{C} -objects B , the functor $(-) \otimes B$ has a right adjoint $B \Rightarrow (-)$. The associated natural isomorphisms $[-]_{\otimes} : \mathcal{C}(A \otimes B, C) \cong \mathcal{C}(A, B \Rightarrow C) : [-]_{\otimes}$ are currying and uncurrying respectively. We call the counit of this adjunction $\text{app} : (A \Rightarrow B) \otimes A \rightarrow B$. Note that for a morphism $g : A \otimes B \rightarrow C$, it is the case that the morphism $A \otimes B \xrightarrow{[g]_{\otimes} \otimes \text{id}} (B \Rightarrow C) \otimes B \xrightarrow{\text{app}} C$ is equal to g .

For objects A , B , and C , we define the following morphism k :

$$k = \left(((B \Rightarrow C) \otimes (A \Rightarrow B)) \otimes A \xrightarrow{\cong} (B \Rightarrow C) \otimes ((A \Rightarrow B) \otimes A) \xrightarrow{\text{id} \otimes \text{app}} (B \Rightarrow C) \otimes B \xrightarrow{\text{app}} C \right)$$

We define the ‘composition’ morphism $\text{comp} = [k]_{\otimes} : (B \Rightarrow C) \otimes (A \Rightarrow B) \rightarrow A \Rightarrow C$ and the identity morphism $\text{ident} = [I \otimes A \xrightarrow{\cong} A]_{\otimes} : I \rightarrow (A \Rightarrow A)$. The triple $\langle A \Rightarrow A, \text{comp}, \text{ident} \rangle$ forms a monoid.

5.2 Cayley representation of ‘Kleisli’ Eilenberg–Moore monoids

The codensity monad of a functor $G : \mathcal{D} \rightarrow \mathcal{E}$ is given by the right Kan extension of G along itself $\text{Ran}_G G : \mathcal{E} \rightarrow \mathcal{E}$ (see Mac Lane [23, Ch. X] or Leinster [22]). Using the coend representation of Kan extensions, one can implement the codensity monad of a Haskell functor f as follows:

```
data Cod f a = Cod (forall x. (a -> f x) -> f x)
```

It is known that the codensity monad of a right adjoint is isomorphic to the monad induced by the adjunction. Hinze [14] gives the following example of how one can use this fact to derive a continuation-based implementation of the list monad. First, we can simulate the forgetful functor from the category of monoids using a class constraint:

```
data L1 a = L1 (forall w. (Monoid w) => (a -> w) -> w)
```

Now, instead of relying on instances of the `Monoid` class, one can use the universal monoid of endomorphisms $x \rightarrow x$. A classic result by Cayley states that every monoid can be represented as a submonoid of the universal monoid (this submonoid is called the *Cayley representation*). Thus, we can equivalently define the monad in question as follows:

```
data L2 a = L2 (forall x. (a -> x -> x) -> x -> x)
```

In this section, we give a similar construction to obtain a continuation-based implementation of the list monad transformer. We start with an Eilenberg–Moore monoid of endomorphisms $MA \Rightarrow MA$:

Theorem 15. *Let \mathcal{C} be closed monoidal. Then, the tuple*

$$\langle MA \Rightarrow MA, [p]_{\otimes}, \text{comp}, \text{ident} \rangle,$$

where

$$p = \left(M(MA \Rightarrow MA) \otimes MA \xrightarrow{\tau} M((MA \Rightarrow MA) \otimes MA) \xrightarrow{\text{Mapp}} MMA \xrightarrow{\mu} MA \right),$$

is an Eilenberg–Moore monoid.

Unfortunately, the Eilenberg–Moore monoid defined in Theorem 15 is not universal. For that, we would have to define a morphism $\langle A, a, m, u \rangle \rightarrow \langle MA \Rightarrow MA, [p]_{\otimes}, \text{comp}, \text{ident} \rangle$ for each Eilenberg–Moore monoid $\langle A, a, m, u \rangle$, while it is in general not possible to define a morphism $g : A \rightarrow (MA \Rightarrow MA)$ that is a morphism between Eilenberg–Moore algebras $\langle A, a \rangle$ and $\langle MA \Rightarrow MA, [p]_{\otimes} \rangle$. To make the construction work, we need to slightly restrict the domain of the forgetful functor U .

First, we give some intuition. The universal property of the adjunction $F \dashv U$ described in Theorem 11 is a folding property: given morphisms $a : MA \rightarrow A$, $m : A \otimes A \rightarrow A$, $u : I \rightarrow A$, and $h : B \rightarrow A$, as long as a , m , and u satisfy the conditions given in the definition of Eilenberg–Moore monoids, we obtain a unique coherent *fold*, that is, a morphism $\text{LMT}_M B \rightarrow A$. It could be also understood as ‘running’ or ‘interpreting’ the monadic computation. However, in programming, when we ‘run’ a backtracking computation, we do not interpret it as a value of some type A . Rather, we interpret it as a value in the

base monad, that is, MA . In other words, we fold the structure of the list, but, instead of eliminating the monadic parts using an Eilenberg–Moore algebra $a : MA \rightarrow A$, we accumulate it using the monadic multiplication μ .

Thus, we are interested in Eilenberg–Moore monoids of the shape $\langle MA, \mu, m, u \rangle$, which we call, for the sake of this article, *Kleisli monoids*, referring to the known fact that the full subcategory of the Eilenberg–Moore category of a monad M that consists of algebras of the shape $\langle MA, \mu \rangle$ is equivalent to the Kleisli category of M . We call the full subcategory of \mathbf{EMon}_M that consists of Kleisli monoids \mathbf{KlMon}_M . The restriction of the forgetful functor $U : \mathbf{EMon}_M \rightarrow \mathcal{C}$ to \mathbf{KlMon}_M is dubbed $U_{\mathbf{Kl}} : \mathbf{KlMon}_M \rightarrow \mathcal{C}$.

A useful observation is that free Eilenberg–Moore monoids defined in Theorem 11 are also Kleisli monoids. This means that $U_{\mathbf{Kl}}$ has a left adjoint $F_{\mathbf{Kl}}$ defined in the same way as F , and that the monad induced by $F_{\mathbf{Kl}} \dashv U_{\mathbf{Kl}}$ is the same monad as the one induced by $F \dashv U$, that is, \mathbf{LMT}_M . Therefore, the monad \mathbf{LMT}_M is also isomorphic to the codensity monad of $U_{\mathbf{Kl}}$. This way, it is enough for our purposes to find a Cayley representation of Kleisli monoids, not necessarily all Eilenberg–Moore monoids. The Eilenberg–Moore monoid $MA \Rightarrow MA$ from Theorem 15, although not a Kleisli monoid itself, is universal for Kleisli monoids:

Theorem 16. *For each Kleisli monoid $\langle MA, \mu, m, u \rangle$, the morphism $[m]_{\otimes} : MA \rightarrow (MA \Rightarrow MA)$ has the following properties:*

- it is an Eilenberg–Moore monoid morphism $\langle MA, \mu, m, u \rangle \rightarrow \langle MA \Rightarrow MA, [p]_{\otimes}, \text{comp}, \text{ident} \rangle$,
- it is a split monomorphism in \mathcal{C} , that is, there exists a morphism $r : (MA \Rightarrow MA) \rightarrow MA$ in \mathcal{C} such that $r \cdot [m]_{\otimes} = \text{id}$.

Using the codensity monad for this representation yields the following monad transformer:

```
type Backtr m a = forall x. (a -> m x -> m x) -> m x -> m x
```

It is the same monad transformer as obtained, although using different methods, by Hinze [13].

6 Revisiting the ‘effects-first’ transformer for commutative monads

Now, we revisit the commutative-monad transformer $m [a]$ known from the `mt1` library in Haskell. We call it a ‘commutative-monad transformer’, as it is a monad if and only if the transformed monad is commutative (see, for example, Mulry [25]). In this section, we derive its continuation-based isomorph, recreating the steps for the \mathbf{LMT}_M monad presented in previous sections. We assume that \mathcal{C} is symmetric closed, and that M is commutative.

6.1 Background: symmetric monoidal categories and commutative monads

A monoidal category is *symmetric* if it is equipped with a natural isomorphism $A \otimes B \xrightarrow{\text{sym}} B \otimes A$ such that sym is an involution (that is, $\text{sym} \cdot \text{sym} = \text{id}$) and the following diagrams commute:

$$\begin{array}{ccc}
 I \otimes A & \xrightarrow{\text{sym}} & A \otimes I \\
 \searrow \cong & & \downarrow \cong \\
 & & A
 \end{array}
 \qquad
 \begin{array}{ccccc}
 (A \otimes B) \otimes C & \xrightarrow{\text{sym} \otimes \text{id}} & (B \otimes A) \otimes C & \xrightarrow{\cong} & B \otimes (A \otimes C) \\
 \downarrow \cong & & & & \downarrow \text{id} \otimes \text{sym} \\
 A \otimes (B \otimes C) & \xrightarrow{\text{sym}} & (B \otimes C) \otimes A & \xrightarrow{\cong} & B \otimes (C \otimes A)
 \end{array}$$

In a symmetric monoidal category, we define a *left strength* for a strong monad M :

$$\tau' = \left(A \otimes MB \xrightarrow{\text{sym}} MB \otimes A \xrightarrow{\tau} M(B \otimes A) \xrightarrow{M\text{sym}} M(A \otimes B) \right)$$

One can show that the appropriate mirror images of the diagrams for a strong endofunctor and a strong monad commute for τ' . A monad is *commutative* if it is equipped both with a (right) strength and a left strength, and the following diagram commutes for all objects A and B :

$$\begin{array}{ccccc} MA \otimes MB & \xrightarrow{\tau'} & M(MA \otimes B) & \xrightarrow{M\tau} & MM(A \otimes B) \\ \downarrow \tau & & & & \downarrow \mu \\ M(A \otimes MB) & \xrightarrow{M\tau'} & MM(A \otimes B) & \xrightarrow{\mu} & M(A \otimes B) \end{array}$$

If \mathcal{C} is a closed symmetric monoidal category and M is commutative, we define the Kleisli composition. First, for all objects A , B , and C , consider the following morphism:

$$\begin{aligned} w = \left(((B \Rightarrow MC) \otimes (A \Rightarrow MB)) \otimes A \xrightarrow{\cong} (B \Rightarrow MC) \otimes ((A \Rightarrow MB) \otimes A) \xrightarrow{\text{id} \otimes \text{app}} (B \Rightarrow MC) \otimes MB \right. \\ \left. \xrightarrow{\tau'} M((B \Rightarrow MC) \otimes B) \xrightarrow{M\text{app}} MMC \xrightarrow{\mu} MC \right) \end{aligned}$$

We define the composition of Kleisli morphisms as $\text{kcomp} = [w]_{\otimes} : (B \Rightarrow MC) \otimes (A \Rightarrow MB) \rightarrow A \Rightarrow MC$, and the identity as $[I \otimes A \xrightarrow{\cong} A \xrightarrow{\eta} MA]_{\otimes} : I \rightarrow A \Rightarrow MA$. The triple $\langle A \Rightarrow MA, \text{kcomp}, \text{kident} \rangle$ is a monoid.

6.2 Symmetric Eilenberg–Moore monoids

We now describe how the monad $\mathfrak{m} \ [a]$ arises as a composition of adjoint functors. This is not a new construction, so we skip the proofs. Consider the category **Mon** of monoids in a symmetric monoidal category \mathcal{C} . Assume that the obvious forgetful functor $U_{\mathbf{Mon}} : \mathbf{Mon} \rightarrow \mathcal{C}$ has a left adjoint $F_{\mathbf{Mon}} : \mathcal{C} \rightarrow \mathbf{Mon}$. The induced monad $U_{\mathbf{Mon}}F_{\mathbf{Mon}}$ is the list monad. Now, given a commutative monad M on \mathcal{C} , we define a monad \overline{M} on **Mon** (in fact, a lifting in the sense of Beck [8]). It is given as follows:

$$\begin{aligned} \overline{M}\langle A, m, u \rangle &= \langle MA, MA \otimes MA \xrightarrow{\tau} M(A \otimes MA) \xrightarrow{M\tau'} MM(A \otimes A) \xrightarrow{\mu m} MA, I \xrightarrow{u} A \xrightarrow{\eta} MA \rangle \\ \overline{M}f &= Mf \\ \mu^{\overline{M}} &= \mu \quad \eta^{\overline{M}} = \eta \end{aligned}$$

Let $F_{\overline{M}} \dashv U_{\overline{M}}$ be the Eilenberg–Moore adjunction of \overline{M} . Since adjoint functors compose, we obtain an adjunction $F_{\overline{M}}F_{\mathbf{Mon}} \dashv U_{\mathbf{Mon}}U_{\overline{M}}$. The induced monad $U_{\mathbf{Mon}}U_{\overline{M}}F_{\overline{M}}F_{\mathbf{Mon}}$ corresponds to the Haskell monad $\mathfrak{m} \ [a]$. Here, we call this monad CLT_M (‘commutative list transformer’).

Now, we take a closer look at the Eilenberg–Moore category of \overline{M} . Each object consists of a pair $\langle \langle A, m, u \rangle, a \rangle$, where $\langle A, m, u \rangle$ is a monoid, and a has the Eilenberg–Moore property, that is, $a \cdot Ma = a \cdot \mu$ and $a \cdot \eta = \text{id}$ (since the monadic structure of \overline{M} is identical to the monadic structure of M). Note that a is a morphism in **Mon**, so it preserves the monoid structure. That is, the following two diagrams commute:

$$\begin{array}{ccccc} MA \otimes MA & \xrightarrow{a \otimes a} & A \otimes A & & \\ \downarrow \tau & & \downarrow m & & \\ M(A \otimes MA) & \xrightarrow{M\tau'} & MM(A \otimes A) & \xrightarrow{\mu m} & MA \xrightarrow{a} A \end{array} \quad (1)$$

$$\begin{array}{ccccc}
 I & \xrightarrow{u} & A & \xrightarrow{\eta} & MA \\
 \downarrow u & & & \nearrow a & \\
 A & & & &
 \end{array} \tag{2}$$

Note that the diagram (2) commutes for all a with the Eilenberg–Moore property. This, together with some rearranging of the elements of the tuples, leads us to the following equivalent definition of algebras for \overline{M} :

Definition 17. Let M be a commutative monad on a symmetric monoidal category \mathcal{C} . A *symmetric Eilenberg–Moore M -monoid* is a tuple $\langle A, a, m, u \rangle$, such that:

- $\langle A, a \rangle$ is an Eilenberg–Moore algebra,
- $\langle A, m, u \rangle$ is a monoid,
- coherence: the diagram (1) commutes.

A morphism between two symmetric Eilenberg–Moore M -monoids is given by a \mathcal{C} -morphism that is both a morphism between the Eilenberg–Moore algebra parts and the monoid parts. We call the category of symmetric Eilenberg–Moore M -monoids **SEMMon** $_M$.

The name is justified by the following theorem:

Theorem 18. *Every symmetric Eilenberg–Moore M -monoid is an Eilenberg–Moore M -monoid.*

Remark 19. One could also imagine a ‘twisted’ definition of Eilenberg–Moore monoids that uses τ' instead of τ , and a coherence condition that equates the two corresponding morphisms $A \otimes MA \rightarrow A$. The proof of Theorem 18 can be easily adapted to state that every symmetric Eilenberg–Moore monoid is a ‘twisted’ Eilenberg–Moore monoid. Additionally, one can prove that a quadruple that is both an Eilenberg–Moore monoid and a ‘twisted’ Eilenberg–Moore monoid for a commutative monad M is necessarily a symmetric Eilenberg–Moore monoid.

Since the definition of symmetric Eilenberg–Moore monoids is a simple rearrangement of the definition of algebras for CLT_M , the adjunction $F_{\overline{M}}F_{\text{Mon}} \dashv U_{\text{Mon}}U_{\overline{M}}$ gives us that the obvious forgetful functor $U_{\text{SEMMon}} : \text{SEMMon}_M \rightarrow \mathcal{C}$ has a left adjoint F_{SEMMon} , and that the induced monad is equal to CLT_M . Although a composition of two monadic adjunctions is not always monadic, it is so in this case (it follows from a general theorem of Beck about algebras for composite monads [8, Proposition 2]):

Theorem 20. *The adjunction $F_{\text{SEMMon}} \dashv U_{\text{SEMMon}}$ is strictly monadic. This entails that **SEMMon** $_M$ is isomorphic to the category of algebras for CLT_M .*

6.3 Endomorphism representation and a continuation-based implementation

Now, assume that \mathcal{C} is a closed symmetric monoidal category. We define a symmetric Eilenberg–Moore monoid of Kleisli endomorphisms, that is, objects $A \Rightarrow MA$:

Theorem 21. *Let \mathcal{C} be a symmetric closed monoidal category, and M be a commutative monad on \mathcal{C} . Then, the tuple*

$$\langle A \Rightarrow MA, [q]_{\otimes}, \text{kcomp}, \text{kident} \rangle,$$

where

$$q = \left(M(A \Rightarrow MA) \otimes A \xrightarrow{\tau} M((A \Rightarrow MA) \otimes A) \xrightarrow{\text{Mapp}} MMA \xrightarrow{\mu} MA \right),$$

is a symmetric Eilenberg–Moore monoid.

It is left to prove that the monoid $A \Rightarrow MA$ is universal for the sufficient subcategory of symmetric Eilenberg–Moore monoids:

Theorem 22. *For each symmetric Eilenberg–Moore monoid of the shape $\langle MA, \mu, m, u \rangle$, the morphism $[s]_{\otimes} : MA \rightarrow (A \Rightarrow MA)$, where*

$$s = \left(MA \otimes A \xrightarrow{\text{id} \otimes \eta} MA \otimes MA \xrightarrow{m} MA \right),$$

has the following properties:

- *it is a morphism $\langle MA, \mu, m, u \rangle \rightarrow \langle A \Rightarrow MA, [q]_{\otimes}, \text{kcomp}, \text{kident} \rangle$ between symmetric Eilenberg–Moore monoids,*
- *it is a split monomorphism in \mathcal{C} , that is, there exists a morphism $r : (A \Rightarrow MA) \rightarrow MA$ in \mathcal{C} such that $r \cdot [s]_{\otimes} = \text{id}$.*

The codensity monad that uses the representation above can be encoded in Haskell as follows:

```
data CLT m a = CLT (forall x. (a -> x -> m x) -> x -> m x)
```

Intuitively, this type represents folds over a list-like structure. Folding a single element can produce some effects in the monad m , but it does not depend on the effects produced by previous elements (it can depend on the values though). The nil of the list (that is, the failure continuation) does not produce monadic effects on its own.

7 Discussion

Equations similar to the conditions in the definition of Eilenberg–Moore monoids were previously discussed by Hinze [13], although in a different setting, that is, as equations between Haskell expressions. Jaskelioff and Moggi [18] suggest that an equational theory like the one discussed in Section 3 induces the list monad transformer, but they leave this without a proof. Wand and Vaillancourt [31] use logical relations to compare two metalanguages with backtracking: one based on streams, and the other on a two-continuation monad. Eilenberg–Moore algebras of the resumption monad are also known as *F*- and *M*-algebras. They were used by Filinski and Støvring [10] (and later by Atkey *et al.* [3, 4]) to model data structures that interleave pure data and effects.

In eager languages, the bare list monad is rarely used as a basis of backtracking computations, since the entire list structure is always computed upfront. Thus, in ML-like languages, one uses the type of lazy lists, which produce elements on demand. It can be implemented using the Haskell syntax as follows:

```
data LazyList a = LazyList (() -> Maybe (a, LazyList a))
```

Given a value of this type, one can force the next step of the computation by supplying the unit value $()$, and only then the structure is evaluated. This is nothing else than the list monad transformer (‘done right’) applied to the reader monad $() \rightarrow a$.

Some languages provide separate primitives for inductive and coinductive data. From the point of view of semantics, it means that the language supports types given by initial algebras and final coalgebras separately. It is an interesting challenge for future work to describe the ‘coinductive’ list monad transformer, given by carriers of final coalgebras. In such a case, the free monad becomes the *free completely iterative monad* introduced by Aczel *et al.* [1], and the resumption monad becomes the *coinductive resumption monad* described by Piróg and Gibbons [26]. The universal properties of both constructions

are similar to those of their inductive counterparts, but considerably more complicated (see Adámek *et al.* [2] for the case of the free completely iterative monad).

Another challenge for future work is to extend the current development with control operators, such as Prolog's *cut* or fair disjunction. These features can be found, for example, in Kiselyov *et al.*'s implementation [20]. We hope that such control structures can be obtained using the methods described in this paper.

Acknowledgements

I would like to thank Tom Schrijvers for his remarks on an early draft of this paper, and the anonymous reviewers for their detailed comments and helpful suggestions.

References

- [1] Peter Aczel, Jiří Adámek, Stefan Milius & Jiří Velebil (2003): *Infinite trees and completely iterative theories: a coalgebraic view*. *Theoretical Computer Science* 300(1-3), p. 145, doi:10.1016/S0304-3975(02)00728-4.
- [2] Jiří Adámek, Stefan Milius & Jiří Velebil (2006): *Elgot Algebras*. *Logical Methods in Computer Science* 2(5), doi:10.2168/LMCS-2(5:4)2006.
- [3] Robert Atkey, Neil Ghani, Bart Jacobs & Patricia Johann (2012): *Fibrational Induction Meets Effects*. In Lars Birkedal, editor: *Foundations of Software Science and Computational Structures, Lecture Notes in Computer Science* 7213, Springer, pp. 42–57, doi:10.1007/978-3-642-28729-9_3.
- [4] Robert Atkey & Patricia Johann (2015): *Interleaving data and effects*. *Journal of Functional Programming* 25, doi:10.1017/S0956796815000209.
- [5] Roland Carl Backhouse, Marcel Bijsterveld, Rik van Geldrop & Jaap van der Woude (1995): *Categorical Fixed Point Calculus*. In: *Category Theory and Computer Science*, p. 159179, doi:10.1007/3-540-60164-3_25.
- [6] Michael Barr & Charles Wells (1985): *Toposes, Triples, and Theories*. Grundlehren der mathematischen Wissenschaften, Springer-Verlag, New York. Available at <http://www.tac.mta.ca/tac/reprints/articles/12/tr12abs.html>.
- [7] Jonathan M. Beck (1967): *Triples, Algebras and Cohomology*. Ph.D. thesis, Columbia University. Available at <http://www.tac.mta.ca/tac/reprints/articles/2/tr2abs.html>.
- [8] Jonathan M. Beck (1969): *Distributive laws*. In B. Eckmann, editor: *Seminar on Triples and Categorical Homology Theory, Lecture Notes in Mathematics* 80, Springer Berlin Heidelberg, pp. 119–140, doi:10.1007/BFb0083084. Available at <http://www.tac.mta.ca/tac/reprints/articles/18/tr18abs.html>.
- [9] Richard S. Bird (2006): *A program to solve Sudoku (functional pearl)*. *Journal of Functional Programming* 16(6), pp. 671–679, doi:10.1017/S0956796806006058.
- [10] Andrzej Filinski & Kristian Støvring (2007): *Inductive Reasoning About Effectful Data Types*. *SIGPLAN Not.* 42(9), pp. 97–110, doi:10.1145/1291220.1291168.
- [11] Marcelo Fiore (2013): *An equational metalogic for monadic equational systems*. *Theory and Applications of Categories* 27(18), pp. 465–492. Available at <http://www.tac.mta.ca/tac/volumes/27/18/27-18abs.html>.
- [12] Marcelo P. Fiore, Gordon D. Plotkin & Daniele Turi (1999): *Abstract syntax and variable binding*. In: *14th Annual IEEE Symposium on Logic in Computer Science, Trento, Italy, July 2-5, 1999*, IEEE Computer Society, pp. 193–202, doi:10.1109/LICS.1999.782615.

- [13] Ralf Hinze (2000): *Deriving backtracking monad transformers (functional pearl)*. In Martin Odersky & Philip Wadler, editors: *Proceedings of the Fifth ACM SIGPLAN International Conference on Functional Programming (ICFP '00)*, Montreal, Canada, September 18-21, 2000., ACM, pp. 186–197, doi:10.1145/351240.351258.
- [14] Ralf Hinze (2012): *Kan extensions for program optimisation or: Art and Dan explain an old trick*. In Jeremy Gibbons & Pablo Nogueira, editors: *Mathematics of Program Construction - 11th International Conference, MPC 2012, Madrid, Spain, June 25-27, 2012. Proceedings*, Lecture Notes in Computer Science 7342, Springer, pp. 324–362, doi:10.1007/978-3-642-31113-0_16.
- [15] John Hughes (1995): *The design of a pretty-printing Library*. In Johan Jeuring & Erik Meijer, editors: *Advanced Functional Programming, First International Spring School on Advanced Functional Programming Techniques*, Båstad, Sweden, May 24-30, 1995, Tutorial Text, Lecture Notes in Computer Science 925, Springer, pp. 53–96, doi:10.1007/3-540-59451-5_3.
- [16] Martin Hyland, Gordon D. Plotkin & John Power (2006): *Combining effects: Sum and tensor*. *Theoretical Computer Science* 357(1-3), p. 7099, doi:10.1016/j.tcs.2006.03.013.
- [17] Martin Hyland & John Power (2006): *Discrete Lawvere theories and computational effects*. *Theoretical Computer Science* 366(1-2), pp. 144–162, doi:10.1016/j.tcs.2006.07.007.
- [18] Mauro Jaskielioff & Eugenio Moggi (2010): *Monad transformers as monoid transformers*. *Theoretical Computer Science* 411(5152), p. 44414466, doi:10.1016/j.tcs.2010.09.011.
- [19] Gregory M. Kelly (1980): *A unified treatment of transfinite constructions for free algebras, free monoids, colimits, associated sheaves, and so on*. *Bulletin of the Australian Mathematical Society* 22, pp. 1–83, doi:10.1017/S0004972700006353.
- [20] Oleg Kiselyov, Chung-chieh Shan, Daniel P. Friedman & Amr Sabry (2005): *Backtracking, Interleaving, and Terminating Monad Transformers: (Functional Pearl)*. *SIGPLAN Notices* 40(9), pp. 192–203, doi:10.1145/1090189.1086390.
- [21] Anders Kock (1971): *Closed categories generated by commutative monads*. *Journal of the Australian Mathematical Society* 12, pp. 405–424, doi:10.1017/S1446788700010272.
- [22] Tom Leinster (2013): *Codensity and the ultrafilter monad*. *Theory and Applications of Categories* 28(13), pp. 332–370. Available at <http://www.tac.mta.ca/tac/volumes/28/13/28-13abs.html>.
- [23] Saunders Mac Lane (1998): *Categories for the Working Mathematician*. Graduate Texts in Mathematics, Springer, doi:10.1007/978-1-4757-4721-8.
- [24] Eugenio Moggi (1989): *An Abstract View of Programming Languages*. Technical Report, Edinburgh University. Available at <http://www.lfcs.inf.ed.ac.uk/reports/90/ECS-LFCS-90-113/>.
- [25] Philip Mulry (2013): *Notions of Monad Strength*. In Anindya Banerjee, Olivier Danvy, Kyung-Goo Doh & John Hatcliff, editors: *Semantics, Abstract Interpretation, and Reasoning about Programs: Essays Dedicated to David A. Schmidt on the Occasion of his Sixtieth Birthday*, Manhattan, Kansas, USA, 19-20th September 2013, *Electronic Proceedings in Theoretical Computer Science* 129, Open Publishing Association, pp. 67–83, doi:10.4204/EPTCS.129.6.
- [26] Maciej Piróg & Jeremy Gibbons (2014): *The Coinductive Resumption Monad*. *Electronic Notes in Theoretical Computer Science* 308, pp. 273–288, doi:10.1016/j.entcs.2014.10.015. Mathematical Foundations of Programming Semantics (MFPS XXX).
- [27] Gordon D. Plotkin & John Power (2003): *Algebraic Operations and Generic Effects*. *Applied Categorical Structures* 11(1), pp. 69–94, doi:10.1023/A:1023064908962.
- [28] Exequiel Rivas, Mauro Jaskielioff & Tom Schrijvers (2015): *From monoids to near-semirings: the essence of MonadPlus and alternative*. In Moreno Falaschi & Elvira Albert, editors: *Proceedings of the 17th International Symposium on Principles and Practice of Declarative Programming, Siena, Italy, July 14-16, 2015*, ACM, pp. 196–207, doi:10.1145/2790449.2790514.
- [29] Mauro Jaskielioff, Exequiel Rivas (2014): *Notions of Computation as Monoids*. Available at <http://arxiv.org/abs/1406.4823>. Submitted to the Journal of Functional Programming.

- [30] Philip Wadler (1985): *How to replace failure by a list of Successes: A method for exception handling, backtracking, and pattern matching in lazy functional languages*. In Jean-Pierre Jouannaud, editor: *Functional Programming Languages and Computer Architecture, FPCA 1985, Nancy, France, September 16-19, 1985, Proceedings, Lecture Notes in Computer Science 201*, Springer, pp. 113–128, doi:10.1007/3-540-15975-4_33.
- [31] Mitchell Wand & Dale Vaillancourt (2004): *Relating Models of Backtracking*. *SIGPLAN Notices* 39(9), pp. 54–65, doi:10.1145/1016848.1016861.
- [32] Harvey Wolff (1973): *Monads and monoids on symmetric monoidal closed categories*. *Archiv der Mathematik* 24(1), pp. 113–120, doi:10.1007/BF01228184.

A Proofs

A.1 Lemma 9

The conditions (1) and (2) from Definition 6 are trivial. For (3), we need to show that the following diagram commutes:

$$\begin{array}{ccc}
 G^*A \otimes A & \xrightarrow{[g] \otimes \text{id}} & A \otimes A \\
 \downarrow \tilde{\tau} & & \downarrow m \\
 G^*(A \otimes A) & \xrightarrow{G^*m} G^*A & \xrightarrow{[g]} A
 \end{array}$$

We show that both paths satisfy the universal property of strongly generated free monads, so they are equal. The top-right path:

$$\begin{array}{ccccccc}
 GG^*A \otimes A & \xrightarrow{\tau} & G(G^*A \otimes A) & \xrightarrow{G([g] \otimes \text{id})} & G(A \otimes A) & \xrightarrow{Gm} & GA \\
 \downarrow \text{cons} \otimes \text{id} & \searrow F[g] \otimes \text{id} & & \nearrow \tau & & & \downarrow g \\
 & & GA \otimes A & & & & \\
 & \text{①} & & \text{②} & & \text{③} & \\
 & & \downarrow g \otimes \text{id} & & & & \\
 G^*A \otimes A & \xrightarrow{[g] \otimes \text{id}} & A \otimes A & \xrightarrow{m} & A & & \\
 \uparrow \eta \otimes \text{id} & \nearrow \text{id} & & & & & \\
 A \otimes A & & & & & &
 \end{array}$$

① $[g]$ is an algebra morphism, ② naturality of τ , ③ $\langle A, g, m, u \rangle$ is a G -monoid, ④ universal property of $[-]$.

The left-bottom path:

$$\begin{array}{ccccccc}
 GG^*A \otimes A & \xrightarrow{\tau} & G(G^*A \otimes A) & \xrightarrow{G\tilde{\tau}} & GG^*(A \otimes A) & \xrightarrow{GG^*m} & GG^*A & \xrightarrow{G[g]} & GA \\
 \downarrow \text{cons} & & \downarrow \text{cons} & & \downarrow \text{cons} & & \downarrow \text{cons} & & \downarrow g \\
 G^*A \otimes A & \xrightarrow{\tilde{\tau}} & G^*(A \otimes A) & \xrightarrow{G^*m} & G^*A & \xrightarrow{[g]} & A & & \\
 \uparrow \eta \otimes \text{id} & \nearrow \eta & & & \uparrow \eta & & \uparrow \text{id} & & \\
 A \otimes A & & & & A & & & &
 \end{array}$$

① definition of strongly generated free monad, ② naturality of cons , ③ $\llbracket - \rrbracket$ is a morphism of algebras, ④ properties of strength, ⑤ naturality of η , ⑥ universal property of $\llbracket - \rrbracket$.

A.2 Theorem 11

We split the proof into a number of lemmata. We need to show that F is a functor (Lemma 23), that $\llbracket - \rrbracket$ is natural (Lemma 24), that $\llbracket - \rrbracket$ produces morphisms between Eilenberg–Moore monoids (Lemma 26), and that $\llbracket - \rrbracket$ is an inverse of $\lceil - \rceil$ (Lemma 27).

Lemma 23. *The assignment F is a functor.*

Proof. First, we check that FA is an Eilenberg–Moore monoid, that is, that the three conditions from Definition 6 hold. The first two are obvious. For the third one, consider the following diagram, which is the desired coherence diagram with the definitions of m and u unfolded:

$$\begin{array}{ccc}
 MM(A \ltimes M)^* I \otimes M(A \ltimes M)^* I & \xrightarrow{\mu^M \otimes \text{id}} & M(A \ltimes M)^* I \otimes M(A \ltimes M)^* I \\
 \downarrow \tau^M & & \downarrow \tau^M \\
 M(M(A \ltimes M)^* I \otimes M(A \ltimes M)^* I) & \text{①} & \\
 \downarrow M\tau^M & & \\
 MM((A \ltimes M)^* I \otimes M(A \ltimes M)^* I) & \xrightarrow{\mu^M} & M((A \ltimes M)^* I \otimes M(A \ltimes M)^* I) \\
 \downarrow MM\widetilde{\tau^M} & & \downarrow M\widetilde{\tau^M} \\
 MM(A \ltimes M)^* (I \otimes M(A \ltimes M)^* I) & \text{②} & M(A \ltimes M)^* (I \otimes M(A \ltimes M)^* I) \\
 \downarrow \cong & & \downarrow \cong \\
 MM(A \ltimes M)^* M(A \ltimes M)^* I & \xrightarrow{\mu^M} & M(A \ltimes M)^* M(A \ltimes M)^* I \\
 \downarrow M\mu^{\mathcal{R}} & \text{③} & \downarrow \mu^{\mathcal{R}} \\
 MM(A \ltimes M)^* I & \xrightarrow{\mu^M} & M(A \ltimes M)^* I
 \end{array}$$

① properties of strength, ② naturality of μ^M , ③ $\mu^{\mathcal{R}}$ is defined via a distributive law.

To verify the morphism part, let $f : A \rightarrow B$ be a morphism in \mathcal{C} . It is trivial that Ff is a morphism between Eilenberg–Moore algebra parts of FA and FB , as it amounts to the naturality of μ^M . As for the monoid parts, the preservation of the unit is simply the fact that $M(f \ltimes M)^* I$ is a monad morphism. For

the preservation of the multiplication, consider the following diagram:

$$\begin{array}{ccc}
 M(A \ltimes M)^* I \otimes M(A \ltimes M)^* I & \xrightarrow{M(f \ltimes M)^* I \otimes M(f \ltimes M)^* I} & M(A \ltimes M)^* I \otimes M(A \ltimes M)^* I \\
 \downarrow \tau^M & \textcircled{1} & \downarrow \tau^M \\
 M((A \ltimes M)^* I \otimes M(A \ltimes M)^* I) & \xrightarrow{M((f \ltimes M)^* I \otimes M(f \ltimes M)^* I)} & M((B \ltimes M)^* I \otimes M(B \ltimes M)^* I) \\
 \downarrow M\widetilde{\tau^M} & \textcircled{2} & \downarrow M\widetilde{\tau^M} \\
 M(A \ltimes M)^*(I \otimes M(A \ltimes M)^* I) & \xrightarrow{M(f \ltimes M)^*(\text{id} \otimes M(f \ltimes M)^*)} & M(B \ltimes M)^*(I \otimes M(B \ltimes M)^* I) \\
 \downarrow \cong & \textcircled{3} & \downarrow \cong \\
 M(A \ltimes M)^* M(A \ltimes M)^* I & \xrightarrow{M(f \ltimes M)^* M(f \ltimes M)^*} & M(B \ltimes M)^* M(B \ltimes M)^* I \\
 \downarrow \mu^{\mathcal{A}} & \textcircled{4} & \downarrow \mu^{\mathcal{A}} \\
 M(A \ltimes M)^* I & \xrightarrow{M(f \ltimes M)^*} & M(B \ltimes M)^* I
 \end{array}$$

① naturality of τ , ② the fact that $(f \ltimes M)^*$ preserves strength, and naturality of $\widetilde{\tau}$, ③ naturality of \cong , ④ monad morphism. \square

Lemma 24. *The assignment $\lfloor - \rfloor$ is a natural transformation.*

Proof. Let $f : FA \rightarrow \langle B, b, m^B, u^B \rangle$ and $l : \langle B, b, m^B, u^B \rangle \rightarrow \langle Y, y, m^Y, u^Y \rangle$ be morphisms in \mathbf{EMMon}_M , and $r : X \rightarrow A$ be a morphism in \mathcal{C} . The following diagram commutes, where the top-most path is equal to $\lfloor l \cdot f \cdot Fr \rfloor$, while the bottom-most path is equal to $Ul \cdot \lfloor f \rfloor \cdot r$:

$$\begin{array}{ccccccc}
 & & X \otimes I & \xrightarrow{\text{INF}} & M(X \ltimes M)^* I & \xrightarrow{M(r \ltimes M)^* I} & M(A \ltimes M)^* I \xrightarrow{f} B \xrightarrow{l} Y \\
 & \cong \nearrow & & \searrow r \otimes \text{id} & & & \\
 X & & \textcircled{1} & & \textcircled{2} & & \\
 & \searrow r & A & \xrightarrow{\cong} & A \otimes I & \xrightarrow{\text{INF}} & M(A \ltimes M)^* I
 \end{array}$$

① naturality of \cong , ② definition of INF. \square

Lemma 25. *Given an Eilenberg–Moore M -monoid $\langle B, b, m^B, u^B \rangle$ and a \mathcal{C} -morphism $g : A \rightarrow B$, the tuple $\langle B, A \otimes MB \xrightarrow{g \otimes b} B \otimes B \xrightarrow{m^B} B, m^B, u^B \rangle$ is an $(A \otimes M(-))$ -monoid for $A \otimes M(-)$ understood as a functor with strength given as $(A \otimes MX) \otimes Y \xrightarrow{\cong} A \otimes (MX \otimes Y) \xrightarrow{\text{id} \otimes \tau} A \otimes M(X \otimes Y)$.*

Proof. Since $\langle B, m^B, u^B \rangle$ is a monoid by definition, it is left to check the coherence diagram:

$$\begin{array}{ccccccc}
 (A \otimes MB) \otimes B & \xrightarrow{(g \otimes \text{id}) \otimes \text{id}} & (B \otimes MB) \otimes B & \xrightarrow{(\text{id} \otimes b) \otimes \text{id}} & (B \otimes B) \otimes B & \xrightarrow{m^B \otimes \text{id}} & B \otimes B \\
 \downarrow \cong & \textcircled{1} & \downarrow \cong & \textcircled{2} & \downarrow \cong & & \downarrow m^B \\
 A \otimes (MB \otimes B) & \xrightarrow{g \otimes (\text{id} \otimes \text{id})} & B \otimes (MB \otimes B) & \xrightarrow{\text{id} \otimes (b \otimes \text{id})} & B \otimes (B \otimes B) & \textcircled{3} & \\
 \downarrow \text{id} \otimes \tau & & \downarrow \text{id} \otimes \tau & & \downarrow \text{id} \otimes m^B & & \\
 A \otimes M(B \otimes B) & \textcircled{4} & B \otimes M(B \otimes B) & \textcircled{5} & & & \\
 \downarrow \text{id} \otimes Mm^B & & \downarrow \text{id} \otimes Mm^B & & & & \\
 A \otimes MB & \xrightarrow{g \otimes \text{id}} & B \otimes MB & \xrightarrow{\text{id} \otimes b} & B \otimes B & \xrightarrow{m^B} & B
 \end{array}$$

① and ② naturality of \cong , ③ associativity of m^B , ④ \otimes is a bifunctor, ⑤ coherence of $\langle B, b, m^B, u^B \rangle$. \square

Lemma 26. For a \mathcal{C} -morphism $g : A \rightarrow U\langle B, b, m^B, u^B \rangle$, the morphism $\lceil g \rceil$ is an \mathbf{EMMon}_M -morphism of the type $FA \rightarrow \langle B, b, m^B, u^B \rangle$.

Proof. The fact that $\lceil g \rceil$ is a morphism between the Eilenberg–Moore algebra parts follows from the following diagram, where the right-most edge is equal to $\lceil g \rceil$ unfolded as in Lemma 5(1):

$$\begin{array}{ccccc}
 MM(A \ltimes M)^* I & \xrightarrow{\mu^M} & M(A \ltimes M)^* I & & \\
 \downarrow MM(A \ltimes M)^* u^B & \textcircled{1} & \downarrow M(A \ltimes M)^* u^B & & \\
 MM(A \ltimes M)^* B & \xrightarrow{\mu^M} & M(A \ltimes M)^* B & & \\
 \downarrow & \textcircled{2} & \downarrow & & \\
 \downarrow M\langle b, m^B \cdot (g \otimes \text{id}), \text{id} \rangle & \swarrow MM\llbracket m^B \cdot (g \otimes b) \rrbracket & \searrow M\llbracket m^B \cdot (g \otimes b) \rrbracket & & \\
 & MMB & \xrightarrow{\mu^M} & MB & \\
 \downarrow Mb & \textcircled{5} & \downarrow b & \textcircled{4} & \\
 MB & \xrightarrow{b} & B & &
 \end{array}$$

① and ② naturality of μ^M , ③ and ④ Lemma 5(2), ⑤ b has the Eilenberg–Moore property.

The fact that $\lceil g \rceil$ commutes with monoid multiplication follows from the following diagram, in which

the left-most edge is the definition of m (that is, the multiplication of FA):

$$\begin{array}{ccccc}
 M(A \ltimes M)^* I \otimes M(A \ltimes M)^* I & \xrightarrow{\text{id} \otimes [g]} & M(A \ltimes M)^* I \otimes B & \xrightarrow{[g] \otimes \text{id}} & B \otimes B \\
 \downarrow \tau & \textcircled{1} & \downarrow \tau & & \downarrow m^B \\
 M((A \ltimes M)^* I \otimes M(A \ltimes M)^* I) & \xrightarrow{M(\text{id} \otimes [g])} & M((A \ltimes M)^* I \otimes B) & & \textcircled{5} \\
 \downarrow M\tilde{\tau} & \textcircled{2} & \downarrow M\tilde{\tau} & & \\
 M(A \ltimes M)^*(I \otimes M(A \ltimes M)^* I) & \xrightarrow{M(A \ltimes M)^*(\text{id} \otimes [g])} & M(A \ltimes M)^*(I \otimes B) & & \\
 \downarrow \cong & \textcircled{3} & \downarrow \cong & & \\
 M(A \ltimes M)^* M(A \ltimes M)^* I & \xrightarrow{M(A \ltimes M)^*[g]} & M(A \ltimes M)^* B & & \\
 \downarrow \mu^{\mathcal{R}} & \textcircled{4} & & \searrow \langle\langle b, m^B \cdot (g \otimes \text{id}), \text{id} \rangle\rangle & \\
 M(A \ltimes M)^* I & \xrightarrow{[g]} & & & B
 \end{array} \quad (3)$$

① naturality of τ , ② naturality of $\tilde{\tau}$, ③ naturality of \cong , ④ see below, ⑤ see below.

To verify that the square ④ above commutes, we unfold $[g]$ as in Lemma 5(1). The desired diagram is then as follows:

$$\begin{array}{ccccc}
 M(A \ltimes M)^* M(A \ltimes M)^* I & \xrightarrow{M(A \ltimes M)^* M(A \ltimes M)^* u^B} & M(A \ltimes M)^* M(A \ltimes M)^* B & \xrightarrow{M(A \ltimes M)^* \langle\langle b, m^B \cdot (g \otimes \text{id}), \text{id} \rangle\rangle} & M(A \ltimes M)^* B \\
 \downarrow \mu^{\mathcal{R}} & \textcircled{1} & \downarrow \mu^{\mathcal{R}} & \textcircled{2} & \downarrow \langle\langle b, m^B \cdot (g \otimes \text{id}), \text{id} \rangle\rangle \\
 M(A \ltimes M)^* I & \xrightarrow{M(A \ltimes M)^* u^B} & M(A \ltimes M)^* B & \xrightarrow{\langle\langle b, m^B \cdot (g \otimes \text{id}), \text{id} \rangle\rangle} & B
 \end{array}$$

① naturality of $\mu^{\mathcal{R}}$, ② $\langle\langle -, -, \text{id} \rangle\rangle$ has the Eilenberg–Moore property (Lemma 5(2)).

Below, we detail ⑤ from the diagram (3):

$$\begin{array}{ccccc}
 M(A \ltimes M)^* I \otimes B & \xrightarrow{M(A \ltimes M)^* u^B \otimes \text{id}} & M(A \ltimes M)^* B \otimes B & \xrightarrow{\langle\langle b, m^B \cdot (g \otimes \text{id}), \text{id} \rangle\rangle \otimes \text{id}} & B \otimes B \\
 \downarrow \tau & \textcircled{1} & \downarrow \tau & \textcircled{2} & \downarrow m^B \\
 M((A \ltimes M)^* I \otimes B) & \xrightarrow{M((A \ltimes M)^* u^B \otimes \text{id})} & M((A \ltimes M)^* B \otimes B) & \xrightarrow{M(\llbracket m^B \cdot (g \otimes b) \rrbracket \otimes \text{id})} & M(B \otimes B) \\
 \downarrow M\tilde{\tau} & \textcircled{5} & \downarrow M\tilde{\tau} & \textcircled{6} & \downarrow Mm^B \\
 M(A \ltimes M)^*(I \otimes B) & \xrightarrow{M(A \ltimes M)^*(u^B \otimes \text{id})} & M(A \ltimes M)^*(B \otimes B) & \xrightarrow{M\llbracket m^B \cdot (g \otimes b) \rrbracket} & MB \\
 \downarrow \cong & \textcircled{7} & \downarrow \cong & \textcircled{8} & \downarrow b \\
 M(A \ltimes M)^* B & \xrightarrow{M(A \ltimes M)^* m^B} & & & B
 \end{array}$$

we unfold the definition of $\lfloor f \rfloor$ in $m^B \cdot (\lfloor f \rfloor \otimes \text{id})$ (the bottom edge):

$$\begin{array}{c}
 A \otimes M(A \ltimes M)^* I \xrightarrow{\mu^{\mathcal{R}} \cdot \text{INF}} M(A \ltimes M)^* I \\
 \downarrow \text{id} \otimes f \quad \searrow \cong \otimes \text{id} \quad \textcircled{1} \quad \text{INF} \otimes \text{id} \quad \nearrow m \\
 (A \otimes I) \otimes M(A \ltimes M)^* I \xrightarrow{\text{INF} \otimes \text{id}} M(A \ltimes M)^* I \otimes M(A \ltimes M)^* I \xrightarrow{f \otimes f} B \otimes B \xrightarrow{m^B} B \\
 \downarrow \cong \otimes \text{id} \quad \textcircled{2} \quad \text{INF} \otimes \text{id} \quad \textcircled{3} \quad f \otimes \text{id} \quad \downarrow f \\
 A \otimes B \xrightarrow{\cong \otimes \text{id}} (A \otimes I) \otimes B \xrightarrow{\text{INF} \otimes \text{id}} M(A \ltimes M)^* I \otimes B \xrightarrow{f \otimes \text{id}} B \otimes B \xrightarrow{m^B} B
 \end{array}$$

① see below, ② \otimes is a bifunctor, ③ f is a monoid morphism (as a morphism in \mathbf{EMon}_M).

Below, we detail ① from the diagram above. We unfold the definitions of INF and m.

$$\begin{array}{c}
 A \otimes M(A \ltimes M)^* I \xrightarrow{\text{id} \otimes \eta^M} A \otimes MM(A \ltimes M)^* I \xrightarrow{\text{emb}} (A \ltimes M)^* M(A \ltimes M)^* I \xrightarrow{\eta^M} M(A \ltimes M)^* M(A \ltimes M)^* I \\
 \downarrow \cong \otimes \text{id} \quad \textcircled{1} \quad \text{id} \otimes \cong \quad \downarrow \mu^{\mathcal{R}} \\
 (A \otimes I) \otimes M(A \ltimes M)^* I \xrightarrow{\cong} A \otimes (I \otimes M(A \ltimes M)^* I) \xrightarrow{\text{id} \otimes (\eta^M \otimes \text{id})} A \otimes (MI \otimes M(A \ltimes M)^* I) \xrightarrow{\text{id} \otimes \tau} A \otimes M(I \otimes M(A \ltimes M)^* I) \\
 \downarrow (\text{id} \otimes \eta^M) \otimes \text{id} \quad \textcircled{2} \quad \downarrow \text{id} \otimes (\eta^M \otimes \text{id}) \quad \textcircled{5} \quad \downarrow \text{id} \otimes \eta^M \quad \textcircled{6} \\
 (A \otimes MI) \otimes M(A \ltimes M)^* I \xrightarrow{\cong} A \otimes (MI \otimes M(A \ltimes M)^* I) \xrightarrow{\text{id} \otimes \tau} A \otimes M(I \otimes M(A \ltimes M)^* I) \xrightarrow{\text{emb}} (A \ltimes M)^* (I \otimes M(A \ltimes M)^* I) \\
 \downarrow \text{emb} \otimes \text{id} \quad \textcircled{3} \quad \downarrow \text{id} \otimes \tau \quad \textcircled{4} \quad \downarrow \eta^M \quad \textcircled{7} \quad \downarrow \tau \quad \textcircled{8} \\
 (A \ltimes M)^* I \otimes M(A \ltimes M)^* I \xrightarrow{\tilde{\tau}} (A \ltimes M)^* (I \otimes M(A \ltimes M)^* I) \xrightarrow{\eta^M} M(A \ltimes M)^* (I \otimes M(A \ltimes M)^* I) \\
 \downarrow \eta^M \otimes \text{id} \quad \textcircled{7} \quad \downarrow \eta^M \quad \textcircled{8} \quad \downarrow M\tilde{\tau} \\
 M(A \ltimes M)^* I \otimes M(A \ltimes M)^* I \xrightarrow{\tau} M((A \ltimes M)^* I \otimes M(A \ltimes M)^* I)
 \end{array}$$

① monoidal category, ② naturality of \cong , ③ the fact that $(A \ltimes M)^*$ is strongly generated and the definition of strength of $A \ltimes M$, ④ properties of strength, ⑤ naturality of η^M and emb , ⑥ naturality of η^M , ⑦ properties of strength, ⑧ naturality of η^M .

It is left to show that the following diagram commutes:

$$\begin{array}{ccc}
 I & \xrightarrow{\eta^{\mathcal{R}}} & M(A \ltimes M)^* I \\
 & \searrow u^B & \downarrow f \\
 & & B
 \end{array}$$

It commutes, since $\eta^{\mathcal{R}} = u$, and f is a morphism between the monoid parts of FA and $\langle B, b, m^B, u^B \rangle$. \square

A.3 Theorem 13

We use the strict version of Beck’s monadicity theorem [7] (see Mac Lane [23, Sec. VI.7]). Let $h_0, h_1 : \langle A, a, m^A, u^A \rangle \rightarrow \langle B, b, m^B, u^B \rangle$ be a pair of morphisms between Eilenberg–Moore monoids. Assume that $g : B \rightarrow C$ is a split coequaliser of Uh_0 and Uh_1 , that is, the following diagram commutes, and the top and bottom paths are both equal to id :

$$\begin{array}{ccccc} B & \xrightarrow{t} & A & \xrightarrow{h_0} & B \\ g \downarrow & & \downarrow h_1 & & \downarrow g \\ C & \xrightarrow{s} & B & \xrightarrow{g} & C \end{array}$$

By Beck’s theorem, it is enough to show that there exists a unique Eilenberg–Moore monoid $\langle C, c, m^C, u^C \rangle$ such that g is a morphism of Eilenberg–Moore monoids $\langle B, b, m^B, u^B \rangle \rightarrow \langle C, c, m^C, u^C \rangle$ and a coequaliser of h_0 and h_1 .

We notice that h_0 and h_1 are also morphisms of Eilenberg–Moore algebras $\langle A, a \rangle \rightarrow \langle B, b \rangle$. Hence, by the fact that the Eilenberg–Moore adjunction is monadic, there exists a unique Eilenberg–Moore algebra $\langle C, c \rangle$ such that $g : \langle B, b \rangle \rightarrow \langle C, c \rangle$ is a unique coequaliser of h_0 and h_1 understood as morphisms between Eilenberg–Moore algebras. In detail, the algebra $\langle C, c \rangle$ is equal to:

$$\langle C, MC \xrightarrow{Ms} MB \xrightarrow{b} B \xrightarrow{g} C \rangle$$

Similarly, h_0 and h_1 are morphism between monoids $\langle A, m^A, u^A \rangle$ and $\langle B, m^B, u^B \rangle$. Since the adjunction between \mathcal{C} and the category of monoids is monadic, there exists a unique monoid $\langle C, m^C, u^C \rangle$ such that $g : \langle B, m^B, u^B \rangle \rightarrow \langle C, m^C, u^C \rangle$ is a unique coequaliser of h_0 and h_1 understood as morphism between monoids. In detail, the monoid $\langle C, m^C, u^C \rangle$ is given as:

$$\langle C, C \otimes C \xrightarrow{s \otimes s} B \otimes B \xrightarrow{m^B} B \xrightarrow{g} C, I \xrightarrow{u^B} B \xrightarrow{g} C \rangle$$

To check that $\langle C, c, m^C, u^C \rangle$ is an Eilenberg–Moore monoid, we need to verify the coherence condition:

$$\begin{array}{ccccccc} MC \otimes C & \xrightarrow{Ms \otimes \text{id}} & MB \otimes C & \xrightarrow{b \otimes \text{id}} & B \otimes C & \xrightarrow{g \otimes \text{id}} & C \otimes C \\ \tau \downarrow & \textcircled{1} & \downarrow \tau & \textcircled{2} & \downarrow \text{id} \otimes s & \textcircled{4} & \downarrow s \otimes s \\ M(C \otimes C) & \xrightarrow{M(s \otimes \text{id})} & M(B \otimes C) & \xrightarrow{b \otimes \text{id}} & B \otimes B & \xrightarrow{g \otimes g} & B \otimes B \\ M(s \otimes s) \downarrow & \textcircled{5} & \downarrow M(\text{id} \otimes s) & \textcircled{6} & \downarrow m^B & \textcircled{7} & \downarrow m^B \\ M(B \otimes B) & \xrightarrow{Mm^B} & MB & \xrightarrow{b} & B & \xrightarrow{g} & C \\ & & \downarrow Mg & \textcircled{8} & \downarrow b & & \\ & & MC & \xrightarrow{Ms} & MB & \xrightarrow{b} & B \end{array}$$

① and ② naturality of τ , ③ \otimes is a bifunctor, ④ $g \cdot s = \text{id}$, since g is a split coequaliser, ⑤ \otimes is a bifunctor, ⑥ $\langle B, b, m^B, u^B \rangle$ is an Eilenberg–Moore monoid, ⑦ g is a morphism between monoids, ⑧ g is a morphism between Eilenberg–Moore algebras.

Morphisms in \mathbf{EMMon}_M need to be exactly morphisms between Eilenberg–Moore algebras and monoids. Thus, g is a morphism $\langle B, b, m^B, u^B \rangle \rightarrow \langle C, c, m^C, u^C \rangle$ and a coequaliser of h_0 and h_1 .

Since $[-]_{\otimes}$ is an isomorphism, it is enough to show that the $[-]_{\otimes}$ -images of both paths in the diagram are equal. The top-right path:

$$\begin{aligned}
 [\text{comp} \cdot (\lfloor \mu \cdot M\text{app} \cdot \tau \rfloor_{\otimes} \otimes \text{id})]_{\otimes} &= [\text{comp}]_{\otimes} \cdot ((\lfloor \mu \cdot M\text{app} \cdot \tau \rfloor_{\otimes} \otimes \text{id}) \otimes \text{id}) && \text{naturality of } [-]_{\otimes} \\
 &= [\lfloor k \rfloor_{\otimes}]_{\otimes} \cdot ((\lfloor \mu \cdot M\text{app} \cdot \tau \rfloor_{\otimes} \otimes \text{id}) \otimes \text{id}) && \text{definition of comp} \\
 &= k \cdot ((\lfloor \mu \cdot M\text{app} \cdot \tau \rfloor_{\otimes} \otimes \text{id}) \otimes \text{id}) && \text{isomorphism} \\
 (*) &= k \cdot (((\text{id} \Rightarrow \mu) \cdot (\text{id} \Rightarrow M\text{app})) \cdot \lfloor \tau \rfloor_{\otimes}) \otimes \text{id} && \text{naturality of } [-]_{\otimes}
 \end{aligned}$$

The left-bottom path:

$$\begin{aligned}
 \lfloor \mu \cdot M\text{app} \cdot \tau \rfloor_{\otimes} \cdot M\text{app} \cdot \tau &= \lfloor \mu \cdot M\text{app} \cdot \tau \rfloor_{\otimes} \cdot (M\text{app} \otimes \text{id}) \cdot (\tau \otimes \text{id}) && \text{naturality of } [-]_{\otimes} \\
 (**) &= \mu \cdot M\text{app} \cdot \tau \cdot (M\text{app} \otimes \text{id}) \cdot (\tau \otimes \text{id}) && \text{isomorphism}
 \end{aligned}$$

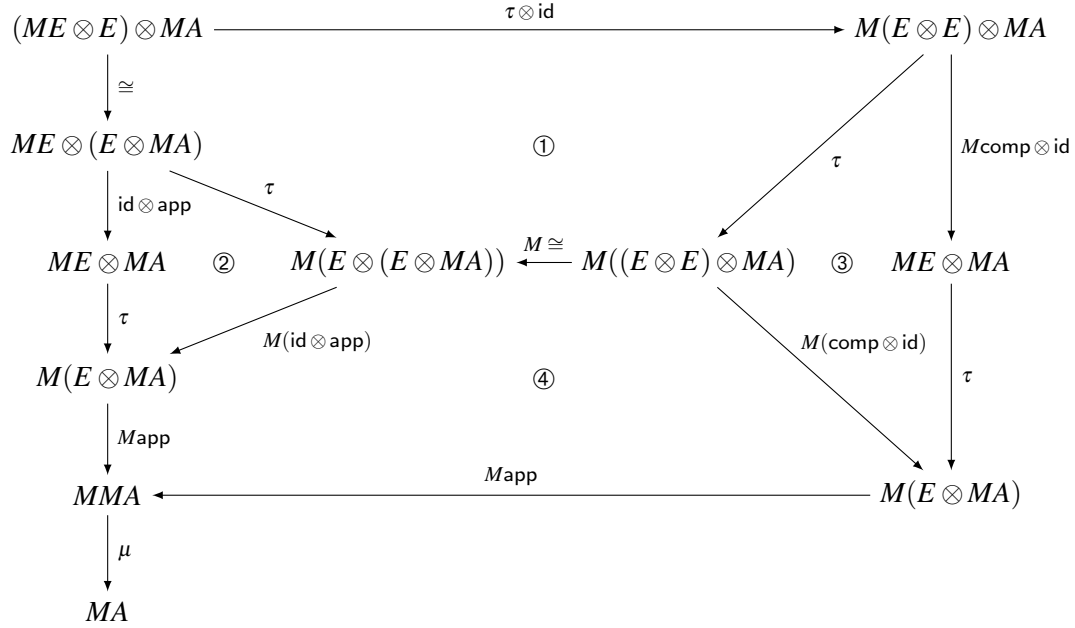
To show that $(*)$ is equal to $(**)$, we split the desired diagram into two. First, consider the following diagram, where the left-bottom path is equal to $(*)$. For brevity, we denote the object $MA \Rightarrow MA$ as E .

$$\begin{array}{ccccc}
 (ME \otimes E) \otimes MA & \xrightarrow{\cong} & ME \otimes (E \otimes MA) & \xrightarrow{\text{id} \otimes \text{app}} & ME \otimes MA \\
 \downarrow (\lfloor \tau \rfloor_{\otimes} \otimes \text{id}) \otimes \text{id} \quad \textcircled{1} & & \downarrow \lfloor \tau \rfloor_{\otimes} \otimes \text{id} & & \downarrow \lfloor \tau \rfloor_{\otimes} \otimes \text{id} \\
 (MA \Rightarrow M(E \otimes MA)) \otimes (E \otimes MA) & \xrightarrow{\text{id} \otimes \text{app}} & (MA \Rightarrow M(E \otimes MA)) \otimes MA & & \\
 \downarrow \cong & \textcircled{2} & \downarrow \text{id} \otimes \text{app} & & \downarrow \tau \\
 ((MA \Rightarrow M(E \otimes MA)) \otimes E) \otimes MA & \xrightarrow{\text{comp} \otimes \text{id}} & (MA \Rightarrow M(E \otimes MA)) \otimes MA & & \\
 \downarrow ((\text{id} \Rightarrow M\text{app}) \otimes \text{id}) \otimes \text{id} & \textcircled{3} & \downarrow (\text{id} \Rightarrow M\text{app}) \otimes \text{id} & & \downarrow \text{app} \\
 ((MA \Rightarrow MMA) \otimes E) \otimes MA & \xrightarrow{\text{comp} \otimes \text{id}} & (MA \Rightarrow MMA) \otimes MA & & M(E \otimes MA) \\
 \downarrow ((\text{id} \Rightarrow \mu) \otimes \text{id}) \otimes \text{id} & \textcircled{4} & \downarrow (\text{id} \Rightarrow \mu) \otimes \text{id} & & \downarrow M\text{app} \\
 (E \otimes E) \otimes MA & \xrightarrow{\text{comp} \otimes \text{id}} & E \otimes MA & & MMA \\
 \downarrow \cong & \textcircled{5} & \downarrow \text{id} \otimes \text{app} & & \downarrow \mu \\
 E \otimes (E \otimes MA) & \xrightarrow{\text{id} \otimes \text{app}} & E \otimes MA & \xrightarrow{\text{app}} & MA
 \end{array}$$

① naturality of \cong , ② \otimes is a bifunctor, ③ and ④ app is the counit of the adjunction, ⑤ and ⑥ naturality of comp , ⑦ naturality of app , ⑧ app is the counit of the adjunction.

Now, consider the following diagram, where the left-most path is equal to the top-right path of the

diagram above, while the path around the perimeter is equal to $(**)$:



① properties of strength, ② and ③ naturality of strength, ④ app is the counit of the adjunction.

A.5 Theorem 16

The fact that $\lfloor m \rfloor_{\otimes}$ is a morphism between monoids follows from the construction of Cayley representation of monoids in monoidal categories [28, 29]. The retraction $r : (MA \Rightarrow MA) \rightarrow MA$ is defined as follows:

$$r = \left((MA \Rightarrow MA) \xrightarrow{\cong} (MA \Rightarrow MA) \otimes I \xrightarrow{\text{id} \otimes u} (MA \Rightarrow MA) \otimes MA \xrightarrow{\text{app}} MA \right)$$

To check that $\lfloor m \rfloor_{\otimes}$ is a morphism between Eilenberg–Moore parts, we want the following diagram to commute:

$$\begin{array}{ccc} MMA & \xrightarrow{M\lfloor m \rfloor_{\otimes}} & M(MA \Rightarrow MA) \\ \downarrow \mu & & \downarrow \lfloor p \rfloor_{\otimes} \\ MA & \xrightarrow{\lfloor m \rfloor_{\otimes}} & MA \Rightarrow MA \end{array}$$

Using the naturality of $\lfloor - \rfloor_{\otimes}$, it is enough to show that $\lfloor p \cdot (M\lfloor m \rfloor_{\otimes} \otimes \text{id}) \rfloor_{\otimes} = \lfloor m \cdot (\mu \otimes \text{id}) \rfloor_{\otimes}$. Thus, it is enough to show that $p \cdot (M\lfloor m \rfloor_{\otimes} \otimes \text{id}) = m \cdot (\mu \otimes \text{id})$. It is detailed in the following diagram, where

the right-most edge is equal to p :

$$\begin{array}{ccc}
 MMA \otimes MA & \xrightarrow{M[m]_{\otimes} \otimes \text{id}} & M(MA \Rightarrow MA) \otimes MA \\
 \downarrow \mu \otimes \text{id} & \searrow \tau & \downarrow \tau \\
 MA \otimes MA & \xrightarrow{m} & MA \\
 & \nearrow M([m]_{\otimes} \otimes \text{id}) & \nearrow Mm \\
 & M(MA \otimes MA) & M((MA \Rightarrow MA) \otimes MA) \\
 & \searrow Mm & \downarrow M\text{app} \\
 & MMA & \downarrow \mu \\
 & & MA
 \end{array}$$

① ② ③

① naturality of τ , ② app is the counit of the adjunction, ③ coherence for $\langle MA, \mu, m, u \rangle$.

To verify that $[m]_{\otimes}$ preserves the unit, we want the following diagram:

$$\begin{array}{ccc}
 I & \xrightarrow{u} & MA \\
 \searrow \text{ident} & & \downarrow [m]_{\otimes} \\
 & & MA \Rightarrow MA
 \end{array}$$

Using the naturality of $[-]_{\otimes}$, it is enough to show that $[m \cdot (u \otimes \text{id})]_{\otimes} = \text{ident}$. Thus, it is enough to show that $m \cdot (u \otimes \text{id}) = [\text{ident}]_{\otimes} = (I \otimes MA \xrightarrow{\cong} MA)$. This follows from the fact that $\langle MA, m, u \rangle$ is a monoid.

A.6 Theorem 18

Except for the assumption that M is commutative, the only difference between the definition of symmetric Eilenberg–Moore monoids and the definition of regular Eilenberg–Moore monoids is in the coherence conditions. Thus, given a symmetric Eilenberg–Moore monoid $\langle A, a, m, u \rangle$, it is enough to show that it satisfies the coherence condition for Eilenberg–Moore monoids:

$$\begin{array}{ccccc}
 MA \otimes A & & \xrightarrow{a \otimes \text{id}} & & A \otimes A \\
 \downarrow \tau & \searrow \text{id} \otimes \eta & & \searrow a \otimes a & \downarrow m \\
 MA \otimes MA & \xrightarrow{\tau} & M(A \otimes MA) & & \\
 \downarrow \eta & \searrow \tau' & \downarrow M\tau' & & \\
 M(MA \otimes A) & \xrightarrow{M\tau} & MM(A \otimes A) & \xrightarrow{\mu} & M(A \otimes A) \xrightarrow{Mm} MA \\
 \downarrow M\tau & \searrow \eta & \downarrow \mu & \nearrow \text{id} & \downarrow a \\
 MM(A \otimes A) & \xrightarrow{\mu} & M(A \otimes A) & \xrightarrow{Mm} & MA \\
 \downarrow \eta & \nearrow \text{id} & \downarrow Mm & \nearrow a & \\
 M(A \otimes A) & \xrightarrow{Mm} & MA & \xrightarrow{a} & A
 \end{array}$$

① ② ③ ④ ⑤ ⑥ ⑦

- ① naturality of η , ② properties of strength, ③ a has the Eilenberg–Moore property, ④ M is commutative, ⑤ coherence for symmetric Eilenberg–Moore monoids, ⑥ monad laws, ⑦ identity.

A.7 Theorem 21

The fact that $[q]_{\otimes}$ has the Eilenberg–Moore property follows from exactly the same reasoning as for $[p]_{\otimes}$ in the proof of Theorem 15.

It is left to verify the coherence condition. It is shown by the following diagram (and subsequent two diagrams, each detailing a part of the previous one), where E stands for $A \Rightarrow MA$:

$$\begin{array}{c}
 \begin{array}{ccccc}
 (ME \otimes ME) \otimes A & \xrightarrow{([q]_{\otimes} \otimes [q]_{\otimes}) \otimes \text{id}} & (E \otimes E) \otimes A & & \\
 \downarrow \tau \otimes \text{id} & \searrow \cong & \downarrow \cong & & \\
 M(E \otimes ME) \otimes A & & ME \otimes (ME \otimes A) & \xrightarrow{[q]_{\otimes} \otimes ([q]_{\otimes} \otimes \text{id})} & E \otimes (E \otimes A) \\
 \downarrow M\tau' \otimes \text{id} & \text{①} & \downarrow \text{id} \otimes \tau & \text{③} & \downarrow \text{id} \otimes \text{app} \\
 MM(E \otimes E) \otimes A & \xrightarrow{\mu \otimes \text{id}} & M(E \otimes E) \otimes A & & ME \otimes M(E \otimes A) \\
 \downarrow MM\text{kcomp} \otimes \text{id} & & \downarrow \tau & \text{④} & \downarrow \text{id} \otimes M\text{app} \\
 MME \otimes A & & M((E \otimes E) \otimes A) & \xrightarrow{M(\text{id} \otimes \text{app})} & M(E \otimes MA) \\
 \downarrow \mu \otimes \text{id} & & \downarrow M(\text{kcomp} \otimes \text{id}) & \text{⑥} & \downarrow M\tau' \\
 ME \otimes A & \xrightarrow{\tau} & M(E \otimes A) & \xrightarrow{M\text{app}} & MMA \\
 & & \downarrow M\mu & & \downarrow \mu \\
 & & MMA & \xrightarrow{\mu} & MA
 \end{array}
 \end{array}$$

- ① detailed below, ② naturality of \cong , ③ app is the counit of the adjunction, ④ definition of q , ⑤ naturality of μ and τ , ⑥ app is the counit of the adjunction.

$$\begin{array}{c}
(ME \otimes ME) \otimes A \xrightarrow{\cong} ME \otimes (ME \otimes A) \xrightarrow{id \otimes \tau} ME \otimes M(E \otimes A) \xrightarrow{id \otimes Mapp} ME \otimes MMA \\
\downarrow \tau \otimes id \quad \textcircled{1} \quad \downarrow \tau \quad \textcircled{2} \quad \downarrow \tau \quad \textcircled{3} \quad \swarrow id \otimes \mu \quad \downarrow [q] \otimes \mu \\
M(E \otimes ME) \otimes A \xrightarrow{\tau} M((E \otimes ME) \otimes A) \xrightarrow{M(id \otimes \tau)} M(E \otimes M(E \otimes A)) \xrightarrow{M(id \otimes Mapp)} M(E \otimes MMA) \\
\downarrow M\tau' \otimes id \quad \textcircled{4} \quad \downarrow M(\tau' \otimes id) \quad \textcircled{5} \quad \downarrow M(id \otimes \mu) \quad \downarrow \tau \quad \downarrow M\tau' \quad \downarrow M\tau \quad \downarrow Mq \quad \downarrow Mapp \\
MM(E \otimes E) \otimes A \xrightarrow{\tau} M(M(E \otimes E) \otimes A) \xrightarrow{M\tau} MM(E \otimes MA) \xrightarrow{MMapp} MMMA \xrightarrow{M\mu} MMA \xrightarrow{\mu} MA \\
\downarrow \mu \otimes id \quad \textcircled{10} \quad \downarrow M\tau \quad \downarrow M\tau' \quad \downarrow \mu \quad \downarrow \mu \quad \downarrow \mu \quad \downarrow \mu \quad \downarrow \mu \quad \downarrow \mu \\
M(E \otimes E) \otimes A \xrightarrow{\tau} M((E \otimes E) \otimes A) \xrightarrow{M\tau} MM(E \otimes (E \otimes A)) \xrightarrow{MM\tau'} MMM(E \otimes A) \xrightarrow{MMMapp} MMMMA \xrightarrow{M\mu} MMMA \xrightarrow{\mu} MMA \xrightarrow{\mu} MA \\
\downarrow \tau \quad \downarrow \mu \quad \downarrow M\tau' \quad \downarrow \mu \quad \downarrow \mu \quad \downarrow \mu \quad \downarrow \mu \quad \downarrow \mu \quad \downarrow \mu \\
M((E \otimes E) \otimes A) \xrightarrow{M\tau} MM(E \otimes (E \otimes A)) \xrightarrow{MM\tau'} MMM(E \otimes A) \xrightarrow{MMMapp} MMMMA \xrightarrow{M\mu} MMMA \xrightarrow{\mu} MMA \xrightarrow{\mu} MA \\
\downarrow M\tau' \quad \downarrow \mu \quad \downarrow \mu \quad \downarrow \mu \quad \downarrow \mu \quad \downarrow \mu \quad \downarrow \mu \quad \downarrow \mu \quad \downarrow \mu \\
M(E \otimes (E \otimes A)) \xrightarrow{M\tau'} MM(E \otimes MA) \xrightarrow{MM\tau'} MMM(E \otimes A) \xrightarrow{MMMapp} MMMMA \xrightarrow{M\mu} MMMA \xrightarrow{\mu} MMA \xrightarrow{\mu} MA
\end{array}$$

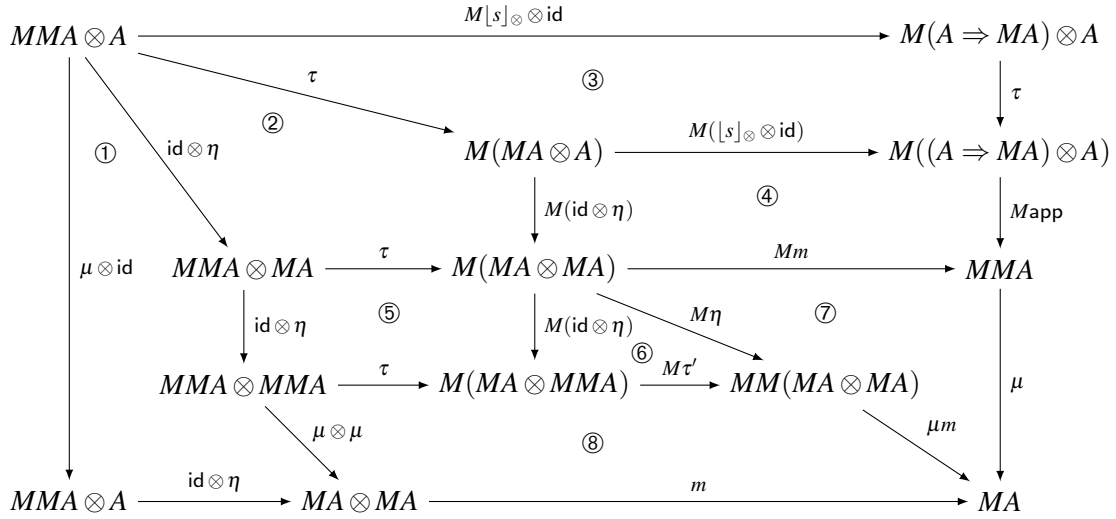
① properties of strength, ② ③ ④ naturality of τ , ⑤ detailed below, ⑥ M is commutative, ⑦ naturality of τ' , ⑧ app is the counit of the adjunction, ⑨ definition of q , ⑩ properties of strength, ① and ② naturality of μ , ③ monad laws.

$$\begin{array}{c}
M((E \otimes ME) \otimes A) \xrightarrow{M\cong} M(E \otimes (ME \otimes A)) \xrightarrow{M(id \otimes \tau)} M(E \otimes M(E \otimes A)) \xrightarrow{M(id \otimes Mapp)} M(E \otimes MMA) \\
\downarrow M(\tau' \otimes id) \quad \textcircled{1} \quad \downarrow M\tau' \quad \textcircled{2} \quad \downarrow M\tau' \quad \textcircled{3} \quad \downarrow M(id \otimes \mu) \\
M(M(E \otimes E) \otimes A) \xrightarrow{M\tau} MM((E \otimes E) \otimes A) \xrightarrow{MM\tau'} MMM(E \otimes A) \xrightarrow{MMMapp} MMMMA \xrightarrow{M\mu} MMMA \xrightarrow{\mu} MMA \xrightarrow{\mu} MA \\
\downarrow M\tau \quad \downarrow MM\tau' \quad \downarrow \mu \quad \downarrow \mu \quad \downarrow \mu \quad \downarrow \mu \quad \downarrow \mu \quad \downarrow \mu \\
MM((E \otimes E) \otimes A) \xrightarrow{MM\tau} M(M(E \otimes E) \otimes A) \xrightarrow{M(M\tau')} M(MM(E \otimes A)) \xrightarrow{M(M\mu)} M(MMA) \xrightarrow{M\mu} MMA \xrightarrow{\mu} MA \\
\downarrow MM\tau' \quad \downarrow M\tau' \quad \downarrow M\mu \quad \downarrow \mu \quad \downarrow \mu \quad \downarrow \mu \quad \downarrow \mu \quad \downarrow \mu \\
MM(E \otimes (E \otimes A)) \xrightarrow{MM\tau'} MMM(E \otimes A) \xrightarrow{MMMapp} MMMMA \xrightarrow{M\mu} MMMA \xrightarrow{\mu} MMA \xrightarrow{\mu} MA \\
\downarrow \mu \quad \downarrow \mu \quad \downarrow \mu \quad \downarrow \mu \quad \downarrow \mu \quad \downarrow \mu \quad \downarrow \mu \quad \downarrow \mu \\
M(E \otimes (E \otimes A)) \xrightarrow{M\tau'} MM(E \otimes MA) \xrightarrow{MM\tau'} MMM(E \otimes A) \xrightarrow{MMMapp} MMMMA \xrightarrow{M\mu} MMMA \xrightarrow{\mu} MMA \xrightarrow{\mu} MA \\
\downarrow M\tau' \quad \downarrow \mu \quad \downarrow \mu \quad \downarrow \mu \quad \downarrow \mu \quad \downarrow \mu \quad \downarrow \mu \quad \downarrow \mu \\
M(E \otimes MA) \xrightarrow{M\tau'} MM(E \otimes A) \xrightarrow{MM\tau'} MMM(E \otimes A) \xrightarrow{MMMapp} MMMMA \xrightarrow{M\mu} MMMA \xrightarrow{\mu} MMA \xrightarrow{\mu} MA
\end{array}$$

① M is a bistrong monad, ② naturality of τ' , ③ properties of strength, ④ monad laws, ⑤ ⑥ ⑦ ⑧ naturality of μ , ⑨ monad laws.

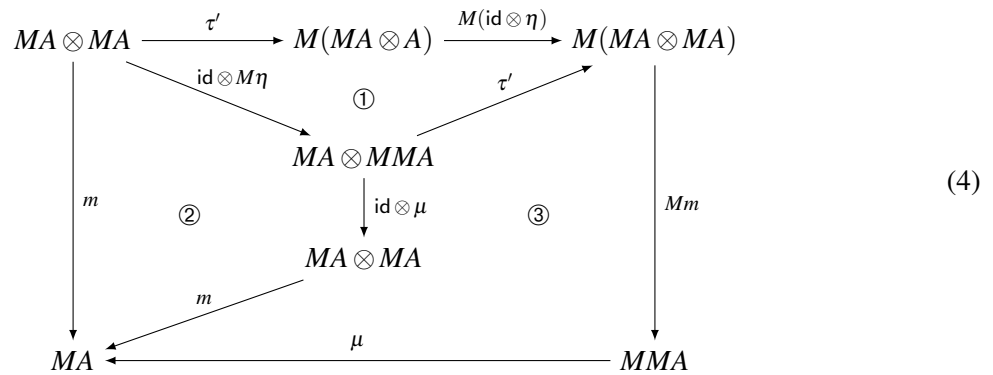
A.8 Theorem 22

We need to show that $[s]_{\otimes}$ is a morphism between Eilenberg–Moore algebras. It is enough to show that the following diagram commutes:



① monad laws, ② and ③ naturality of τ , ④ app is the counit of the adjunction, and the definition of s
 ⑤ naturality of τ , ⑥ properties of strength, ⑦ monad laws, ⑧ coherence for $\langle MA, \mu, m, u \rangle$.

Next, we need to $[s]_{\otimes}$ is a morphism between the monoid parts. It is easy to verify that it preserves the unit. As for the preservation of monoid multiplication, we first show that the following diagram commutes:



① naturality of τ' , ② monad laws, ③ the ‘twisted’ coherence condition for τ' (see Remark 19).

Now, to see that $[s]_{\otimes}$ preserves the monoid multiplication, it is enough to show that the following diagram commutes, in which E stands for $A \Rightarrow MA$:

$$\begin{array}{c}
\begin{array}{ccc}
(MA \otimes MA) \otimes A & \xrightarrow{([s]_\otimes \otimes [s]_\otimes) \otimes \text{id}} & (E \otimes E) \otimes A \\
\downarrow \text{id} \otimes \eta & \textcircled{2} & \downarrow \text{id} \otimes \eta \\
\textcircled{1} \quad (MA \otimes MA) \otimes MA & \xrightarrow{([s]_\otimes \otimes [s]_\otimes) \otimes \text{id}} & (E \otimes E) \otimes MA \\
\downarrow \cong & \textcircled{3} & \downarrow \cong \\
MA \otimes (MA \otimes MA) & \xrightarrow{[s]_\otimes \otimes ([s]_\otimes \otimes \text{id})} & E \otimes (E \otimes MA) \\
\downarrow \text{id} \otimes \tau' & \textcircled{5} & \downarrow \text{id} \otimes \tau' \\
MA \otimes M(MA \otimes A) & \xrightarrow{[s]_\otimes \otimes M([s]_\otimes \otimes \text{id})} & E \otimes M(E \otimes A) \\
\downarrow \text{id} \otimes M(\text{id} \otimes \eta) & \textcircled{7} & \downarrow \text{id} \otimes M\text{app} \\
MA \otimes M(MA \otimes MA) & \xrightarrow{[s]_\otimes \otimes Mm} & E \otimes MMA \\
\downarrow \tau' & \textcircled{9} & \downarrow \tau' \\
M(MA \otimes (MA \otimes MA)) & \xrightarrow{M([s]_\otimes \otimes m)} & M(E \otimes MA) \\
\downarrow M(\text{id} \otimes m) & \textcircled{1} & \downarrow M\tau' \\
M(MA \otimes MA) & \xrightarrow{M([s]_\otimes \otimes \text{id})} & MM(E \otimes A) \\
\downarrow M\tau' & \textcircled{2} & \downarrow \mu \\
MM(MA \otimes A) & \xrightarrow{\mu} & M(MA \otimes A) \\
\downarrow MM(\text{id} \otimes \eta) & \textcircled{5} & \downarrow M(\text{id} \otimes \eta) \\
MM(MA \otimes MA) & \xrightarrow{\mu} & M(MA \otimes MA) \\
\downarrow MMm & \textcircled{7} & \downarrow Mm \\
MMA & \xrightarrow{M\mu} & MMA \\
\downarrow Mm & \textcircled{9} & \downarrow \mu \\
MA \otimes A & \xrightarrow{\text{id} \otimes \eta} & MA \otimes MA \xrightarrow{m} MA
\end{array}
\end{array}$$

The diagram illustrates a complex commutative structure involving various tensor products and monad operations. The nodes are arranged in a grid-like fashion, with arrows representing the transformations between them. The transformations are labeled with mathematical expressions and circled numbers (1 through 10) indicating specific properties or lemmas used in the proof.

- ① see below, ② \otimes is a bifunctor, ③ and ④ naturality of \cong , ⑤ naturality of τ' , ⑥ properties of strength, ⑦ app is the counit of the adjunction, ⑧ naturality of η , ⑨ naturality of τ' , ⑩ properties of strength, ⑪ \otimes is a bifunctor, ⑫ naturality of τ' , ⑬ naturality of η , and monad laws, ⑭ and ⑮ naturality of μ , ⑯ app is the counit of the adjunction, ⑰ naturality of μ , ⑱ monad laws, ⑲ diagram (4).

Below, we detail ① from the diagram above.

$$\begin{array}{ccc}
 (MA \otimes MA) \otimes A & \xrightarrow{m \otimes \text{id}} & MA \otimes A \\
 \downarrow \text{id} \otimes \eta & \text{①} & \downarrow \text{id} \otimes \eta \\
 (MA \otimes MA) \otimes MA & \xrightarrow{m \otimes \text{id}} & MA \otimes MA \\
 \downarrow \cong & \text{②} & \\
 MA \otimes (MA \otimes MA) & \xrightarrow{\text{id} \otimes m} & MA \otimes MA \\
 \downarrow \text{id} \otimes \tau' & & \uparrow \text{id} \otimes \mu \\
 MA \otimes M(MA \otimes A) & \text{③} & \\
 \downarrow \text{id} \otimes M(\text{id} \otimes \eta) & & \downarrow m \\
 MA \otimes M(MA \otimes MA) & \xrightarrow{\text{id} \otimes Mm} & MA \otimes MMA \\
 \downarrow \tau' & \text{⑤} & \downarrow \tau' \\
 M(MA \otimes (MA \otimes MA)) & \xrightarrow{M(\text{id} \otimes m)} & M(MA \otimes MA) \xrightarrow{Mm} MMA \xrightarrow{\mu} MA
 \end{array}$$

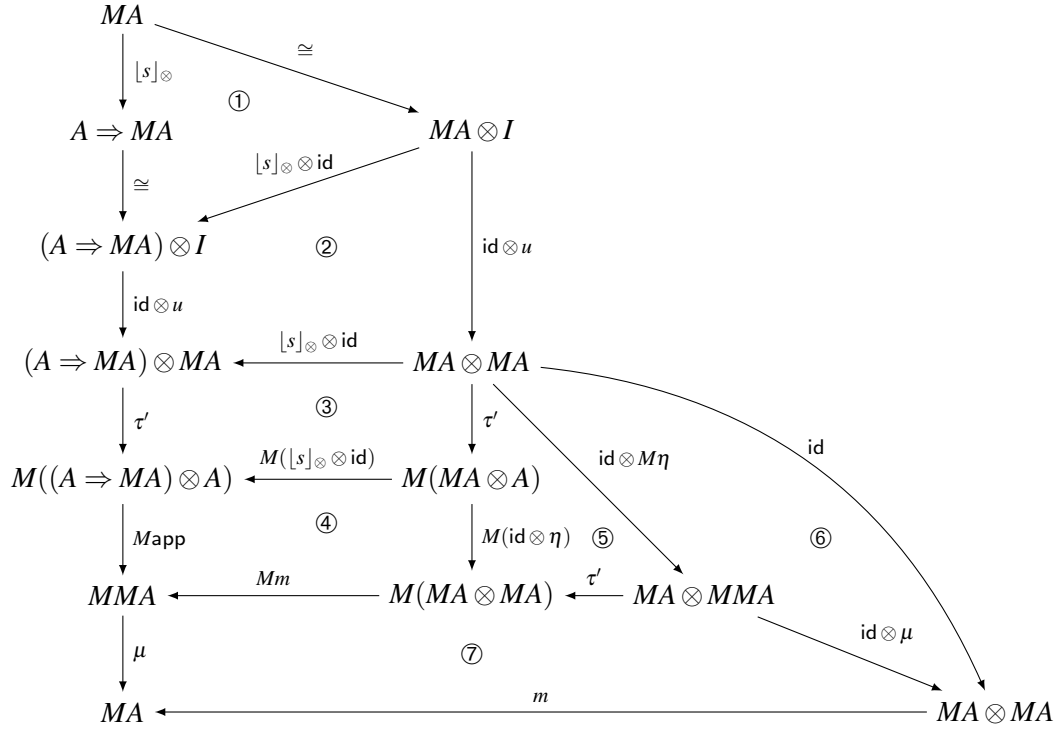
④

- ① \otimes is a bifunctor, ② associativity of monoid multiplication, ③ $(\text{id} \otimes (-))$ -image of the diagram (4), ④ the ‘twisted’ coherence condition for τ' (see Remark 19), ⑤ naturality of τ' .

It is left to show that $[s]_{\otimes}$ is a split mono. The retraction r is given as follows:

$$r = \left((A \Rightarrow MA) \xrightarrow{\cong} (A \Rightarrow MA) \otimes I \xrightarrow{\text{id} \otimes u} (A \Rightarrow MA) \otimes MA \xrightarrow{\tau'} M((A \Rightarrow MA) \otimes A) \xrightarrow{M\text{app}} MMA \xrightarrow{\mu} MA \right)$$

We verify that $r \cdot [s]_{\otimes} = \text{id}$:



① naturality of \cong , ② \otimes is a bifunctor, ③ naturality of τ' , ④ app is the counit of the adjunction, and the definition of s , ⑤ naturality of τ' , ⑥ monad laws, ⑦ the ‘twisted’ coherence condition for τ' (see Remark 19)